THE UNIVERSITY *of* EDINBURGH

THE UNIVERSITY *of* EDINBURGH

Numerical Simulation of Shock Propagation in One and Two Dimensional Domains

Hatice Kursungecmez

Doctor of Philosophy
University of Edinburgh
2015

Declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text.

Hatice Kursungecmez

September 2015, *Edinburgh*

Lay Summary

In 1888, the convergent-divergent nozzle was developed by Gustaf de Laval [1]. This nozzle operates at three different gas speeds i.e., subsonic (less than the speed of sound), sonic (the speed of sound) and supersonic (more than the speed of sound) based on the acceleration of the flow by the contraction of the nozzle. The speed of sound is approximately 768 miles per hour (or equivalently 1,236 kilometers per hour). The variations in the gas speed are created due to the asymmetric hour glass shape of the nozzle. This nozzle is used in steam turbines and as a rocket engine nozzle. In 1947, the first manned aircraft flying at supersonic speed was designed by Bell Aircraft Company and named the XS-1 [1, 2]. Here, "XS" stands for experimental and supersonic [2]. Later on, it was named the X-1 and became one of the many experimental X-vehicles for supersonic flight made by NASA [2]. Since the properties of the flow cannot be visualized by the naked eye, mathematical and numerical studies are needed to help to understand them.

Therefore, a proper understanding of the above physical phenomena and other real world problems, e.g., explosions and blast waves, supersonic flight and the propagation of shock waves in tubes or channels, can be achieved by solving the systems of non-linear conservation laws governing these problems. In general, it is impossible to solve the equations for these complicated flows exactly. To overcome this problem numerical methods have been developed to provide numerical approximations of the true solution of the governing equations. These equations can then be solved using computer programs. In this thesis, the equations of supersonic flow, the compressible Euler equations of gas dynamics, are used to solve for shock propagation in a curved channel. The numerical solutions obtained in this thesis are compared with experimental results and

excellent agreement is found. This shows the utility of the numerical methods used here.

Abstract

The objective of this dissertation is to develop robust and accurate numerical methods for solving the compressible, non-linear Euler equations of gas dynamics in one and two space dimensions. In theory, solutions of the Euler equations can display various characteristics including shock waves, rarefaction waves and contact discontinuities. To capture these features correctly, highly accurate numerical schemes are designed. In this thesis, two different projects have been studied to show the accuracy and utility of these numerical schemes.

Firstly, the compressible, non-linear Euler equations of gas dynamics in one space dimension are considered. Since the non-linear partial differential equations (PDEs) can develop discontinuities (shock waves), the numerical code is designed to obtain stable numerical solutions of the Euler equations in the presence of shocks. Discontinuous solutions are defined in a weak sense, which means that there are many different solutions of the initial value problems of PDEs. To choose the physically relevant solution among the others, the entropy condition was applied to the problem. This condition is then used to derive a bound on the solution in order to satisfy L^2-stability. Also, it provides information on how to add an adequate amount of diffusion to smooth the numerical shock waves. Furthermore, numerical solutions are obtained using far-field and no penetration (wall) boundary conditions. Grid interfaces were also included in these numerical computations.

Secondly, the two dimensional compressible, non-linear Euler equations are considered. These equations are used to obtain numerical solutions for compressible flow in a shock tube with a 90° circular bend for two channels of different curvatures. The cell centered finite volume numerical scheme is employed to achieve these numerical

solutions. The accuracy of this numerical scheme is tested using two different methods. In the first method, manufactured solutions are used to the test the convergence rate of the code. Then, Sod's shock tube test case is implemented into the numerical code to show the correctness of the code in both flow directions. The numerical method is then used to obtain numerical solutions which are compared with experimental data available in the literature. It is found that the numerical solutions are in a good agreement with these experimental results.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my first supervisor, Prof. Noel Smyth, for his patience, motivation and immense knowledge. This has been a great opportunity for me, and I appreciate his encouragement and assistance along the way.

I am sincerely thankful to my second supervisor, Dr.Maximilian Ruffert and Dr.James Maddison who contributed to this dissertation by their knowledge and guidance. In addition, I express my appreciation to Dr.Joan Simon and Dr.Gennady El for being in my committee. Their thoughtful questions and comments were valued greatly. I would also thank Prof. Magnus Svärd for his guidance during the first years of my PhD and his collaboration. Without their precious support, it would not be possible to conduct this work.

I am grateful to Turkish Ministry of Education for their financial support through my PhD study.

Thanks to all of my friends in the University of Edinburgh. I could never have got through without your support and friendship.

Last but not the least, I owe my sincere gratitude to my parents, Meryem and Recep, and my sister, Hande for their endless love and support through the most difficult times in my entire life. Finally, thank you to my dear husband, Dursun for your unwavering support, love and care. I always feel lucky to have you in my life.

Contents

Lay Summary iii

Abstract v

Acknowledgements vi

List of Figures x

List of Tables xvi

1 Introduction 1

1.1 Project One : Shock propagation in one-dimensional space 4

1.2 Project Two : Shock propagation in two-dimensional curved channels . 7

1.3 Layout of the thesis 9

2 Entropy Stable Schemes for the Euler Equations with Boundary Conditions 10

2.1 Weak Solutions and Entropy Condition 12

2.2 Overview of entropy stable schemes for Cauchy Case 20

2.3 Entropy stable schemes for a bounded domain 22

2.3.1 Lax Wendroff Theorem 33

2.3.2 The Inviscid Burgers' Equation 35

2.3.3 The Euler Equations of Gas Dynamics 37

2.3.3.1 Far-field Boundary Conditions 41

2.3.3.2 Wall Boundary Conditions 43

2.3.3.3 Grid Interface Condition 47

2.3.3.4 Diffusion Limiter 51
2.4 Numerical Computations 53
2.4.1 Sod's Shock Tube Test 54
2.4.2 Test for Shock-Entropy Wave Interaction 55
2.4.3 Test for Interaction of Shock and Wall 56
2.4.4 Test for Grid Interface Treatment of Shock 58
2.4.5 A Fourth Order Entropy Stable Scheme 59

3 Method of Manufactured Solutions(MMS) 64
3.1 Implementation of MMS into Compressible Euler Equations of Gas Dynamics in 2D 66
3.1.1 Numerical Formulation 66
3.1.2 Analytical Manufactured Solution of the System of PDEs 67
3.1.3 Source Terms of Conservation of Mass, Momentum, and Energy Equations 68
3.1.4 Verification of Designed Computer Code : Convergence Rate . . 71
3.2 Spatial Discretization Numerical Method 72
3.3 Numerical Simulations and Results 73

4 The Sod's Shock Tube Problem 81
4.1 Introduction 81
4.2 Governing Equations 83
4.2.1 Spatial Discretization 84
4.2.2 Temporal Integration 86
4.2.3 CFL condition 87
4.3 Numerical Simulations and Results 88

5 Shock propagation in curved channels 93
5.1 Experimental Study 93
5.2 The Euler equations of gas dynamics in polar coordinates 96
5.2.1 Spatial Discretization for 2D : CCFVM 99
5.2.2 Temporal Integration for 2D:SSPRK 103

5.2.3 Initial Conditions 104
5.2.4 Boundary Conditions and Ghost Cells 107
5.3 Numerical Simulations and Results 111

6 Conclusions and Future Work **143**

Appendices **148**

A Entropy Conservative Inviscid Burgers' Equation **149**

B The Algorithm of Logarithmic Mean **151**

C Matlab Code for position angle of shock waves **152**

List of Figures

1.1 The first illustration of supersonic flow over a bullet. This figure is taken from [8]. 2

1.2 Sketch of the channels with different radii of curvature [43]. Here, the radii of curvature R_1 and R_2 are of sharp and shallow channels, respectively. $(R_1 < R_2)$ 7

2.1 Numerical solution of ρ with 100 grid points **(a)** at $Time = 2.0$, **(b)** at $Time = 4.0$. 54

2.2 Numerical solution of ρ with 200 grid points **(a)** at $Time = 2.0$, **(b)** at $Time = 4.0$. 55

2.3 Numerical solution of ρ at $Time = 1.8$ **(a)** with 800 grid points, **(b)** with 3000 grid points. 56

2.4 Numerical solution of ρ with 400 grid points **(a)** at $Time = 1.8$, **(b)** at $Time = 2.5$, **(c)** at $Time = 2.8$. 57

2.5 Numerical solution of ρu with 400 grid points **(a)** at $Time = 0.05$, **(b)** at $Time = 0.38$. 58

2.6 Numerical solution of ρu with 400 grid points **(a)** at $Time = 0.5$, **(b)** at $Time = 0.6$. 58

2.7 Numerical solution of ρ at $Time = 1.8$ **(a)** with 200 and 80 grid points for the left and right domains, **(b)** with 80 and 200 grid points for the left and right domains. 59

2.8 Numerical solution of ρ at $Time = 1.8$ **(a)** with 200 grid points for both domains, **(b)** pointwise difference. 60

2.9 Numerical solution of ρu with 200 grid points **(a)** at $Time = 1.8$, **(b)** at $Time = 2.8$. 63

3.1 The dark blue (dashed) line denoted by $(\Delta x = \Delta y)^2$, demonstrates the theoretical error variation for a numerical method of second order of accuracy. 75

3.2 2D MMS analytical solutions; density function ρ(at the top left), momentum function in -x direction ρu(at the top right), momentum function in -y direction, ρv(at the bottom left), energy function E(at the bottom right) 76

3.3 Errors between exact and numerical solutions of density, ρ, according to different mesh sizes, $\Delta x = 0.1$, $\Delta y = 0.1$(at top left), $\Delta x = 0.05$, $\Delta y = 0.05$(at top right), $\Delta x = 0.025$, $\Delta y = 0.025$(at bottom left), $\Delta x = 0.0125$, $\Delta y = 0.0125$(at bottom right) 77

3.4 Errors between exact and numerical solutions of momentum in -x direction, ρu, according to different mesh sizes, $\Delta x = 0.1$, $\Delta y = 0.1$(at top left), $\Delta x = 0.05$, $\Delta y = 0.05$(at top right), $\Delta x = 0.025$, $\Delta y = 0.025$(at bottom left), $\Delta x = 0.0125$, $\Delta y = 0.0125$(at bottom right) 78

3.5 Errors between exact and numerical solutions of momentum in -y direction, $rhov$, according to different mesh sizes, $\Delta x = 0.1$, $\Delta y = 0.1$(at top left), $\Delta x = 0.05$, $\Delta y = 0.05$(at top right), $\Delta x = 0.025$, $\Delta y = 0.025$(at bottom left), $\Delta x = 0.0125$, $\Delta y = 0.0125$(at bottom right) 79

3.6 Errors between exact and numerical solutions of energy, E, according to different mesh sizes, $\Delta x = 0.1$, $\Delta y = 0.1$(at top left), $\Delta x = 0.05$, $\Delta y = 0.05$(at top right), $\Delta x = 0.025$, $\Delta y = 0.025$(at bottom left), $\Delta x = 0.0125$, $\Delta y = 0.0125$(at bottom right) 80

4.1 Sketch of the regions initially at $t = 0$ and after breaking the diaphragm at $t > 0$. This figure is taken from Danaila et al. [84]. 82

4.2 Left part is CFL stable since numerical domain of dependence includes physical domain of dependence, $D_{\mathbf{P}} \subset D_{\mathbf{N}}$. However, right part is unstable since this condition is contrary to CFL stability condition, $D_{\mathbf{P}} \not\subset D_{\mathbf{N}}$. Here $D_{\mathbf{P}}$ is physical domain (orange area), $D_{\mathbf{N}}$ is numerical domain (green area). This figure is taken from Rezzolla et al. [88]. 88

4.3 The initial values of density and pressure at the right and left sides of the diaphragm which is located in the middle of the domain at $x = 5\ m$ at time $t = 0\ s$. 91

4.4 Comparison of numerical and exact solutions for a) Density function, ρ b) Momentum function, ρu c) Mach number, u/a, where $a = \sqrt{\frac{\gamma p}{\rho}}$ is speed of sound d) Velocity function, u e) Pressure, p v.s. distance along the wall in the x direction at $t = 0.0061\ s$. They are stated for initial conditions in equations (4.1)-(4.3). 92

5.1 Sketch of radii of inner, outer walls and length of curvature for channels. Here L refers to the width of the channel. 94

5.2 Sketch of the traveling shock wave in air towards the end of the sharp tube (Channel 1) with a 90° bend. This figure is taken from [42]. 95

5.3 Sketch of the traveling shock wave in air along the midportion of the shallow tube (Channel 2) with a 90° bend at $M_0 = 2.9$. This figure is taken from [42]. 96

5.4 Sketch of the traveling distance of the shock waves along the inner and outer walls. Here, θ_{is} and θ_{os} are position angles of inner and outer wall shock waves. 98

5.5 The sketch of the domains for interior scheme. 101

5.6 Sketch of numerically implemented boundary conditions at the beginning, at the end, as well as the side walls of the channel. 108

5.7 The sketch of the extended domains with one ghost cells (grey dashed lines) around of entire domains. 109

5.8 Illustration of the effect of outflow and no-penatration wall boundary conditions for the sharp bend for the initial Mach number $M_0 = 2.7$. . . 110

5.9 Numerical evolution of θ velocity u_θ for initial Mach number, $M_0 = 1.7$. (channel 1) 119

5.10 Numerical evolution of density for initial Mach number, $M_0 = 1.7$. (channel 1) 120

5.11 Numerical evolution of pressure for initial Mach number, $M_0 = 1.7$. (channel 1) 121

5.12 Initial Mach number 1.7 for channel 1. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively. 122

5.13 Initial Mach number 1.7 for channel 1. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively. 123

5.14 Initial Mach number 2.1 for channel 1. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively. 124

5.15 Initial Mach number 2.1 for channel 1. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively. 125

5.16 Numerical evolution of θ velocity u_θ for initial Mach number, $M_0 = 2.7$. (channel 1) 126

5.17 Numerical evolution of density for initial Mach number, $M_0 = 2.7$. (channel 1) 127

5.18 Numerical evolution of pressure for initial Mach number, $M_0 = 2.7$. (channel 1) 128

5.19 Initial Mach number 2.7 for channel 1. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively. 129

5.20 Initial Mach number 2.7 for channel 1. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively. 130

5.21 Numerical evolution of the θ velocity u_θ for initial Mach number, $M_0 =$ 1.2. (channel 2) 131

5.22 Numerical evolution of density for initial Mach number, $M_0 = 1.2.$ (channel 2) 132

5.23 Numerical evolution of pressure for initial Mach number, $M_0 = 1.2.$ (channel 2) 133

5.24 Initial Mach number 1.2 for channel 2. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively. 134

5.25 Initial Mach number 1.2 for channel 2. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively. 135

5.26 Initial Mach number 1.9 for channel 2. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively. 136

5.27 Initial Mach number 1.9 for channel 2. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively. 137

5.28 Numerical evolution of θ velocity u_θ for initial Mach number, $M_0 = 2.9$. (channel 2) 138
5.29 Numerical evolution of density for initial Mach number, $M_0 = 2.9$. (channel 2) 139
5.30 Numerical evolution of pressure for initial Mach number, $M_0 = 2.9$. (channel 2) 140
5.31 Initial Mach number 2.9 for channel 2. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively. 141
5.32 Initial Mach number 2.9 for channel 2. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively. 142

6.1 The physical and computational domains for the experimental study. . . 146

List of Tables

3.1 L^2 error norm and convergence rate for second order finite volume method 74

Chapter 1

Introduction

The detailed investigation of solutions of the nonlinear partial differential equations (PDEs) governing compressible flow plays a crucial role in gaining a deeper understanding of many real world phenomena. The first real concern for compressible flow commenced with the advent of high-speed flow. The analysis of such requires a careful analysis of the density variation in the flow because of the high pressure forces created in the flow-field [1]. This was first considered in Laval's convergent-divergent steam nozzle for marine use towards the end of 19th century. A proper understanding of supersonic flow became vital due to the advent of the jet engine in the World War Two, which made supersonic flight possible [3]. This understanding was also vital for a proper understanding of blast waves, in particular during the development of nuclear weapons at Los Alamos during the Second World War. The first manned successful supersonic airplane Bell XS-1 broke the sound barrier and became supersonic on the 14th October, 1947 [1]. These studies of supersonic flow provoked detailed studies related to the speed of sound, shock waves (discontinuities), flows at different Mach numbers and the resolution and interaction of shock waves [4, 5]. The speed of sound, which is a thermodynamic property, was first calculated by the French mathematician Laplace, as $a = \sqrt{\gamma p/\rho}$, where $\gamma = \frac{c_p}{c_v}$, p, ρ, c_p and c_v are the ratio of specific heats, the pressure, the density, specific heat at constant pressure and specific heat at constant volume, respectively [1, 6]. The first visualization of shock waves was achieved in Mach's study titled *"Photographische Fixierung der durch Projektile in der Luft eingeleiten Vorgange"* [1, 7] which deals with the supersonic flow passing over a bullet.

In Figure 1.1, the formation of strong and weak shock waves in the head and tail of the bullet, respectively, are demonstrated [8]. Another thermodynamic property, Mach

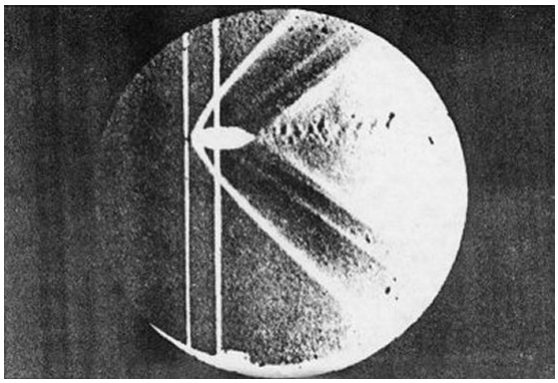


Figure 1.1: The first illustration of supersonic flow over a bullet. This figure is taken from [8].

number, M, was first introduced by Jakop Ackeret [9] and named as the Mach number to honor Ernst Mach for his scientific contributions to supersonic flow [1]. The Mach number is defined as $M = v/a$, where v is the flow velocity and a is the speed of sound. It is also used for the classification of flow regimes, i.e., if the Mach number is $M < 1$, $M = 1$ and $M > 1$, then flow is defined as subsonic, sonic or supersonic, respectively [10]. The relations for the jumps in flow quantities across a shock were derived by Scottish engineer Rankine and French ballistician Hugoniot based on the conservation of mass, momentum and energy [10, 11, 12], and are

$$\rho_1 v_1 = \rho_2 v_2, \quad \text{for Mass,} \tag{1.1}$$

$$p_1 + \rho_1 v_1^2 = p_2 + \rho_2 v_2^2, \quad \text{for Momentum,} \tag{1.2}$$

$$h_1 + \frac{v_1^2}{2} = h_2 + \frac{v_2^2}{2}, \quad \text{for Energy.} \tag{1.3}$$

Here, subscripts $\{_{1,2}\}$ refer to flow quantities ahead and behind the shock wave, respectively, $h = e + p/\rho$ is the enthalpy and e is the internal energy. The use of an artificial diffusion term as introduced in the 1950's by von Neumann and Richtmeyer [13, 14] was critical to stabilize numerical methods for capturing shock waves. This technique prevents nonphysical over and under shoots in the numerical solution and ensures the current entropy jump across the shock [13].

The governing equations formed from the conservation of mass, momentum and

energy are termed *the Euler equations of gas dynamics*. The Euler equations are a prototype example of nonlinear hyperbolic conservation laws. Systems of conservation laws are used to study many real world problems, e.g., modeling flow over an aerofoil or other vehicles, modeling wing flutter, predicting the weather, modeling the dynamics of a single star, traffic flow, flood waves and the propagating of shock waves in tunnels or tubes [10, 15].

Finding analytical solutions of nonlinear PDEs is extremely challenging because of the existence of discontinuities in the form of shock waves. Therefore, numerical methods have been developed to solve these equations [15]. Hence, there is a need to develop and study numerical solution methods in order to approximate the exact solution of such non-linear PDEs. The main classes of such numerical solution methods are finite difference, finite volume, finite element and spectral methods, all of which have different advantages and disadvantages. In this thesis, our aim is to obtain stable numerical solutions converging to the exact solutions of the PDEs using finite difference and finite volume methods.

The historically oldest, most basic and dominantly used numerical method is the finite difference method. The introduction of finite difference methods was given by L.F.Richardson in 1910 [16]. The main purpose of his work was to develop finite difference numerical methods to approximate the solutions of physical problems in irregular domains [16]. The advantage of this method is its easy implementation in a computer program and its potential to allow high order, accurate finite difference schemes in an efficient way. It can also be implemented in multidimensional problems [15]. However, some challenges in terms of solving problems on complex structured geometries can occur using this method since the computational domain for finite difference methods is based on rectangular staggered meshes. This obstacle can be overcome by using curvilinear coordinates for more general domains, which preserves the same order of accuracy as for cartesian coordinates [17]. On the other hand, the finite volume method has advantages for implementation in a computational domain with complex geometries and yields a greater accuracy compared to finite difference methods [17, 18]. Nevertheless, this method becomes computationally expensive due to the complex algorithms for deriving high order schemes. The common fundamental requirement for both methods

is the efficient use of computer resources and memory.

In particular, in the present work we focus on non-linear hyperbolic systems of conservation laws which are time dependent. The stable Runge-Kutta scheme is used in the process of semi-discretization of the governing equations for the temporal discretization. This method was originally developed by the German mathematician Runge [19] and enhanced by Heun [20] and Kutta [21]. Another major advance for the numerical solution of such time dependent hyperbolic conservation laws was introduced by Courant, Friedrichs and Lewy [5]. It was proved that the domain of numerical dependence must include the domain of the mathematical problem [22]. In another words, the speed of the numerical solution must not be faster than the speed of the analytical solution in order to access the required information from the differential equation. This is called CFL condition. It is a necessary stability condition and is generally used for hyperbolic PDEs. A literature review in the field of applied and computational mathematics related to the numerical solution of the problems considered in this thesis is given in the following sections.

1.1 Project One : Shock propagation in one-dimensional space

In the first project, a new numerical approximation for nonlinear hyperbolic systems of conservation laws in one dimensional space is analyzed. The theory of this new numerical method was developed for initial value problems by Osher and Tadmor in the 1980's [23, 24, 25] and such schemes are called entropy stable numerical schemes. This numerical method is based on satisfying the second law of thermodynamics locally and globally for the entire computational domain. The essential point of numerical interpretation of the second law of thermodynamics is that entropy is discretely conserved for smooth regions and increases when the flow encounters shock waves [1]. This requirement is necessary for numerical schemes to explicitly capture physical shock waves and to obtain entropy stable solutions. In our study, Tadmor's [25] framework for constructing entropy conservative and entropy stable schemes is taken into consideration. In [26, 27], Godunov and Mock discovered that when the governing equation is rewritten in entropy variables rather than conservative variables,

the governing system can be symmetrized and one to one mapping can occur between the entropy and conservative variables. Such symmetrization enables the construction of entropy stable schemes by adjusting the amount of artificial numerical diffusion D in such entropy conservative schemes [24]. It was also noted [25] that while there are many suitable entropy variables, $\mathbf{s}^T$, which are generated by the gradient of the chosen convex entropy functions $S_v = \mathbf{s}^T$ for scalar functions, there are fewer for systems of hyperbolic conservation laws. To symmetrize the Euler equations of gas dynamics, a possible family of entropy variables was given by Harten [28]. However, there is only one admissible entropy function used to symmetrize the governing system in our work [29]. This entropy function is the physical entropy function defined by Hughes et al. [30] and symmetrizes the Navier-Stokes equations. The reason to select specifically this physical entropy is that although our main aim is to solve the compressible Euler equations, this is a stepping stone to obtain entropy stable schemes for initial boundary value problems for the compressible Navier-Stokes equations [29]. In [31], Dafermos proved that having obtained an entropy stable solution, an L^2 bound can be implemented on the conservative solution variables. It is worth noting that the novelty of the proposed work in Chapter 2 relies on including boundaries in conservative entropy stable schemes for systems of hyperbolic conservation laws.

In addition, there are different studies dealing with the stability of numerical schemes. The stability of a numerical scheme is primarily studied using the Lax-Ritchmyer theorem [32] for linear problems. This states that a consistent and stable approximation converges to the true solution of a PDE as the computational mesh size goes to zero. The significance of this theorem is that the stability of a consistent numerical scheme implies convergence. Hence, from a numerical perspective, stability becomes the key point to prove convergence. Once the importance of stability was recognized, it has become the subject of more intensive studies for decades. In the following years, another revolutionary stability theorem for nonlinear problems was developed in the work of Lax and Wendroff [33, 34]. The only drawback of this Lax-Wendroff theorem is that it does not guarantee the convergence to a true solution of the PDE. However, it guarantees that the numerical solution is one of the weak solutions of the conservation law. Moreover, the Lax-Wendroff theorem gives confidence in the accuracy of the calculated

weak numerical solution [15].

In [35], a study was performed to derive entropy stable solutions by deriving conservative entropy stable finite difference schemes for initial boundary value problems for the Euler equations. The selected entropy function to symmetrize their system was called the homogeneous entropy function and was introduced in [36]. There are two main drawbacks preventing this entropy being employed in our work. First, the homogeneous entropy is not capable of symmetrizing the Navier-Stokes equations. Second, the boundary entropy inequality condition (see[37]) in [35] was based on an assumption that the diffusion matrix is non-singular. However, this cannot be applied to the Navier-Stokes equations since in this case the diffusion matrix is singular. A number of examples of entropy stable schemes including entropy stable fluxes based on different choices of diffusion matrices have been found in the literature [25, 38, 39].

In Dutt [40], boundary conditions were derived for the compressible Navier-Stokes equations which guarantee an L^2 bound on the solution [40]. The framework of the study by Dutt provides implementable physical boundary conditions for supersonic and subsonic inflow and outflow by scaling the physical entropy function and obtaining the corresponding entropy flux function [29]. The purpose of this applied scaling technique was to force the entropy flux function to change its sign at the inflow or outflow boundary in order to provide a growth limit on the entropy [29]. However, as pointed out in [29], Dutt made an assumption on the density that it was limited to a maximum value in order to obtain entropy stable solutions. However, this is not very convenient from a physical point of view. Therefore, his approach is not used in our study.

In [41], the aim was to obtain high order numerical solutions for initial boundary value problems for the compressible Euler equations. This work was influenced by the previously obtained conservative entropy stable boundary finite difference scheme reported in [35]. The objective of the theory behind deriving high order entropy stable schemes was to unravel the solution structures in complex wave propagation problems, e.g., turbulence, vortices, which cannot be captured by the method of [35] due to its second order accurate scheme for the interior of the computational domain. A more appropriate manner to achieve a least diffusive numerical entropy stable scheme was explained in [41]. This was achieved by constructing novel limiter functions. This

method leads to less diffused solutions within the domain. This is due to the property of limiter functions which can control the amount of artificial diffusion for both smooth and non-smooth regions. It is worth mentioning that the necessity of using a limiter to obtain least diffusive numerical solutions became a standard technique after its suggestion by Rosenblunt [13]. It was pointed out in [13] that artificial diffusion should be added only into the compressed regions but should be neglected for expansion regions since the presence of discontinuity appears in the compressed medium.

1.2 Project Two : Shock propagation in two-dimensional curved channels

The second project is based on an experimental study by Edwards et al. [42], which was performed to investigate the propagation of initially planar shock waves in air along the two different 90° curved channels with rectangular cross sections. The difference between the channels is their different radii of curvatures. This results in sharper and shallower channels (see Figure 1.2). The physical set up of the channels

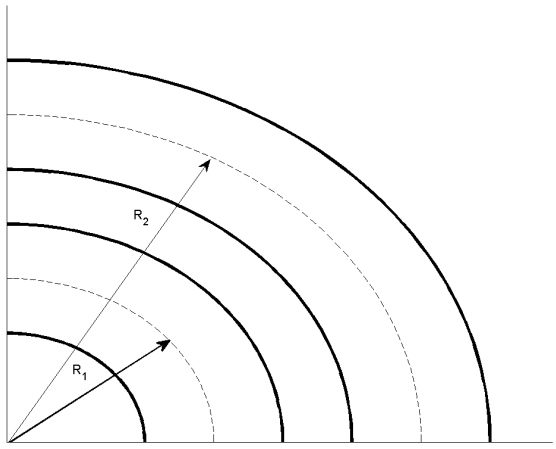


Figure 1.2: Sketch of the channels with different radii of curvature [43]. Here, the radii of curvature R_1 and R_2 are of sharp and shallow channels, respectively. ($R_1 < R_2$)

affects the behavior and velocities of the shock waves while they are propagating towards the downstream of the channels, e.g., the curvature of the shock waves along the sharp channel, regaining the initial planar shock waves towards the end of shallow channel, acceleration on the outer (concave) wall and attenuation on the inner (convex)

wall [42]. The change in the initial shock wave Mach numbers along the convex and concave walls for each channel was experimentally calculated. The experimental results were compared with those obtained using Whitham's geometrical shock theory [10] for the Mach numbers at both walls [42], and by using Courant and Friendrich's 3-shock theory for the concave wall only [44]. Courant and Friedrich's 3-shock theory slightly overestimated the wall Mach numbers of the shock waves on the concave wall, while the predictions of Whitham's geometrical shock theory were in a better agreement with these experimental results [42]. In [43], the numerical propagation of shock waves along the channels was investigated using Whitham's geometrical shock theory [10]. Another study was reported by Schwendeman in 1993 [45] who compared his numerical results with those obtained by using Whitham's theory in the earlier work of [43], and agreement was found. In [46], the numerical implementation of geometrical shock dynamics was extended to three dimensional shock motion and the shock tube experiments of [42] were revisited. In [46], the shock was a two dimensional surface propagating in the shock tube of the experiments of [42]. The Mach numbers at the inner and outer walls were compared with the experimental results. The Mach numbers at the wall were found to be lower than for the one dimensional work of [43]. This was due to the extra deformation of the shock as a two dimensional surface.

It is worth noting that Whitham's geometrical theory provides essential information on the interaction between the geometry and nonlinearity during the propagation of non-linear, unsteady and non-planar shock waves [3, 47, 48]. This breakthrough theory was developed before the availability of the numerical methods able to cope with multidimensional shock motion. However, Whitham's geometrical shock theory does not give the flow field behind the shock wave [10]. To calculate this flow, the gas dynamics equations need to be solved with any shock waves captured [3, 43, 46]. This solution can be found using the numerical work presented in Chapter 5 where the gas dynamic equations are solved numerically based on the two dimensional Euler equations of gas dynamics in a polar coordinate system. The governing Euler equations were solved by applying a cell central finite volume scheme and a strong stability preserving third order Runge-Kutta scheme for the space and time discretizations, respectively [49, 50]. The numerical scheme for the two dimensional governing equations is then verified and

validated. The verification is provided in Chapter 3 by using the method of manufactured solutions for testing whether the developed code is bug-free [51, 52, 53]. In Chapter 4, the validation is carried out by applying Sod's shock tube problem for each direction individually to demonstrate the capability of capturing the physical features of the nonlinear system, i.e., the rarefaction wave, the contact discontinuity and shock waves [54]. After considering the verification and validation of the numerical method, the numerical scheme is implemented for the discretized governing equations. The numerical scheme is then used to study the curved shock tube problem of [42].

1.3 Layout of the thesis

In this thesis, two main projects are studied to obtain numerically stable solutions for the time dependent, compressible hyperbolic conservation laws of gas dynamics in one and two dimensions.

The first project presented in Chapter 2 focuses on deriving robust and accurate entropy stable finite difference schemes for the compressible Euler equations in one space dimension subject to typical far-field and wall boundary conditions [29]. In addition, the interface treatment for connecting two computational domains with different mesh sizes is also presented. Lastly, the theory for the first project is verified by comparing the results from the numerical solutions with those reported previously in different test cases.

The second project involves the investigation of a numerical approach to model the experimental study of [42, 43] for the propagation of shock waves around curved shock tubes with 90° bends. Two dimensional compressible Euler equations in polar coordinates are discretized using cell centered finite volume methods in space and a third order strong stability preserving Runge-Kutta method in time combined with artificial diffusion. Verification and validation procedures are used for the scheme based on the method of manufactured solutions in Chapter 3 and the Sod's shock tube test case in Chapter 4, respectively. After considering the verification and validation proofs of the numerical scheme, we analyze the governing problem and illustrate the accuracy of the numerical scheme by comparing our numerical simulations with the original experimental study of [42] in Chapter 5.

Chapter 2

Entropy Stable Schemes for the Euler Equations with Boundary Conditions

This chapter deals with a new entropy stable numerical scheme for the equations of compressible gas dynamics in one spatial dimension [10]

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -\frac{1}{\rho}\frac{\partial p}{\partial x}, \tag{2.1}$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = 0. \tag{2.2}$$

Here $u(x,t)$ is the fluid velocity, $\rho(x,t)$ is the fluid density and $p(x,t)$ is the fluid pressure. The pressure and density are related through the ideal gas law

$$p = \mathcal{R}\rho T, \tag{2.3}$$

where T is the temperature of the fluid and $\mathcal{R}$ is the gas constant [10]. The theory of entropy stable schemes for initial value problems for hyperbolic conservation laws was developed by Osher [23] and Tadmor [24, 55]. Numerical schemes for hyperbolic initial-boundary value problems have received much less attention, and entropy stable schemes even less. The present work addresses this gap by investigating entropy stable finite difference methods for initial boundary value problems. This chapter forms a

review and in-depth discussion of the work of Svärd and Özcan [29], which follows on from previous work by Svärd [35, 41]. In addition, it gives an expanded account of the proofs of the convergence and stability of the numerical scheme, as outlined in the original work [29].

Derivation of conservation laws

Conservation laws are initially derived in integral form to measure the total amount of conserved physical quantities in a system within a selected domain, $[x_L, x_R]$, such that [15]

$$\int_{x_L}^{x_R} v(x,t)dx = C, \tag{2.4}$$

where $C \in \mathbb{R}$ is a constant. Here, x_L and x_R stand for the spacial positions of the left and right boundaries of the selected domain respectively, and $v(x,t)$ is a physical quantity, e.g., density, momentum or total energy to be conserved. In the case of one-dimensional gas dynamics problem as described in [15], the conservation of physical quantities of the gas filling a tube can be taken into account. There are sources or sinks, so the physical quantities are assumed to be conserved. If there is a gas flowing across the endpoints, x_L and x_R, then any change in the physical quantity is measured by writing equation (2.4) in the following form

$$\frac{\partial}{\partial t}\int_{x_L}^{x_R} v(x,t)dx = g(v(x_L,t)) - g(v(x_R,t)). \tag{2.5}$$

Here, flux $g(v(x_L,t))$ enters into the system from the left boundary at x_L and the flux $g(v(x_R,t))$ leaves the system from the right boundary at x_R. Note that the total variation in the physical quantity $v(x,t)$ in the time interval $t_1 \leq t \leq t_2$ can be obtained by integrating equation (2.5) in time. Hence the *integral form* of the conservation laws becomes

$$\int_{x_L}^{x_R} \left(v(x,t_2) - v(x,t_1)\right) dx = \int_{t_1}^{t_2} \left(g(v(x_L,t)) - g(v(x_R,t))\right) dt. \tag{2.6}$$

To solve the conservation equation using any convenient numerical method, its integral form should first be converted to a differential form. Thus, the physical quantity $v(x,t)$

and the corresponding flux $g(v(x,t))$ are assumed to be differentiable, so that

$$\int_{t_1}^{t_2} \frac{\partial}{\partial t} v(x,t)dt = v(x,t_2) - v(x,t_1) \tag{2.7}$$

and

$$\int_{x_L}^{x_R} \frac{\partial}{\partial x} g(v(x,t))dx = g(v(x_R,t)) - g(v(x_L,t)). \tag{2.8}$$

Substituting equations (2.7) and (2.8) into equation (2.6), the integral form of the conservation law can be rewritten as

$$\int_{t_1}^{t_2} \int_{x_L}^{x_R} \left[\frac{\partial}{\partial t} v(x,t) + \frac{\partial}{\partial x} g(v(x,t)) \right] dxdt = 0. \tag{2.9}$$

Since equation (2.9) must be hold for any selected domain $[x_L, x_R]$ in any time interval $t_1 \leq t \leq t_2$, the integrand must be zero. Therefore, the equation of conservation can be stated in a *differential form* in the following form

$$\frac{\partial}{\partial t} v(x,t) + \frac{\partial}{\partial x} g(v(x,t)) = 0, \tag{2.10}$$

or

$$v(x,t)_t + g(v(x,t))_x = 0. \tag{2.11}$$

Since solutions of nonlinear hyperbolic conservation laws may develop discontinues in finite time, i.e., shock waves, they generally need to be interpreted in a weak form.

2.1 Weak Solutions and Entropy Condition

A finite difference numerical method subject to entropy stability is used to obtain numerical solutions of hyperbolic conservation laws in a bounded domain, i.e., equations of the form

$$v(x,t)_t + g(v(x,t))_x = 0, \qquad x_L \leq x \leq x_R, \quad t \geq 0 \tag{2.12}$$

$$v(x,0) = v^0(x), \qquad t = 0 \tag{2.13}$$

$$H(v(x_L,t), v(x_R,t)) = (\mathbf{b}_L(t), \mathbf{b}_R(t)), \qquad x = x_L, x_R. \tag{2.14}$$

Here, $v(x,t)$ and $g(v(x,t))$ are the vector of unknowns and the corresponding flux vector, respectively.

Since the desired problem to solve is an initial-boundary value problem, initial, $v^0(x)$ and boundary data, $\mathbf{b}_{L,R}(t)$, are required to obtain numerical solutions. In addition, they are bounded and known data, $\mathbf{b}_{L,R} \in L^\infty$. In [29], the initial conditions are bounded and compactly supported, which means that they are equal to zero outside of the selected domain and this yields the solution in a compact set [15]. Also, Dirichlet type boundary conditions are enforced weakly using the boundary operator $H(v(x_{L,R},t))$ to obtain entropy stable numerical solutions. The particular details of the boundary conditions are determined in Section 2.3. Numerical solutions of the governing system (2.12) may blow up in finite time since the derivative of the solution vector can go to infinity, $v(x,t)_x \to \infty$. Thus, numerical solutions of a non-linear system often include singularities, such as shock waves, and they are evaluated in a weak form [56]. Therefore, a compactly supported test function ϕ is chosen and equation (2.12) is multiplied by it,

$$\phi v(x,t)_t + \phi g(v(x,t))_x = 0.$$

After integrating in time and space, this becomes

$$\int_{\mathbb{R}_+} \int_{x_L}^{x_R} \left[\phi v_t + \phi g_x\right] dxdt = 0,$$

where time is positive, $t > 0$, and $t \in \mathbb{R}_+$. Next, applying integration by parts for the time and space derivatives separately yields

$$\int_{\mathbb{R}_+} \int_{x_L}^{x_R} \phi g_x dxdt = \int_{\mathbb{R}_+} \left[\phi g\Big|_{x_L}^{x_R} - \int_{x_L}^{x_R} \phi_x g dx\right] dt$$

and

$$\int_{x_L}^{x_R} \int_{\mathbb{R}_+} \phi v_t dtdx = \int_{x_L}^{x_R} \left[\phi v\Big|_0^\infty - \int_{\mathbb{R}_+} \phi_t v dt\right] dx,$$

on using the compactness property of the test function $\phi(x,\infty)v(x,\infty) \to 0$. Lastly, on combining the above equations, the weak solution of the governing system is obtained

in the form

$$\int_{\mathbb{R}_+}\int_{x_L}^{x_R}(v\phi_t + g\phi_x)\,dxdt + \int_{x_L}^{x_R}\phi(x,0)v(x,0)dx$$
$$= -\int_{\mathbb{R}_+}\left[\phi(x_L,t)g(x_L,t) - \phi(x_R,t)g(x_R,t)\right]dt. \tag{2.15}$$

However, there is more than one weak solution to approximate the solution of the governing system. Since weak solutions are not unique, the extra auxiliary entropy condition is required to select the physically relevant weak solutions of the system [56]. Entropy quantifies the disorder in a system according to the second law of thermodynamics. Although, entropy is conserved for smooth solutions, it jumps to a higher value across a discontinuity in the physical system [1]. There are two different descriptions for entropy available in the literature, i.e. physical entropy and mathematical entropy. While the physical entropy increases in the presence of discontinuity in the system according to second law of thermodynamics, the mathematical entropy decreases. The latter is because of the negative sign concorporated in the mathematical entropy. For instance, while the physical entropy function is $S = -v^2/2$ [57], the mathematical entropy function is $S = v^2/2$ for the inviscid Burgers' equation as detailed in Section 2.3.2. However, there is no substantive difference between these two entropies [57]. The mathematical entropy is only used for the inviscid Burgers' equation (Section 2.3.2) while the physical entropy is used in the numerical method developed in the remaining sections of this chapter.

To clarify the idea of using entropy condition to chose physically relevant weak solutions among many other non-unique weak solutions, the inviscid Burgers' equation can be taken into consideration. Let consider the initial value problem for the inviscid Burgers' equation [58]

$$v_t + \left(\frac{v^2}{2}\right)_x = 0, \quad x \in \mathbb{R},\ t \geq 0, \tag{2.16}$$

with an initial condition of

$$v(x,0) = \begin{cases} 0 & \textit{if}\ \ x < 0, \\ \\ 1 & \textit{if}\ \ x > 0 \end{cases} \tag{2.17}$$

There are two possible solutions of equation (2.16) satisfying the initial condition (2.17) [58]. One possible solution is equation (2.18) on either left and right sides of the shock wave at $x = t/2$.

$$v_1(x,t) = \begin{cases} 0 & if \;\; x < t/2, \\ 1 & if \;\; x > t/2 \end{cases} \tag{2.18}$$

It is worth noting that $v_1(x,t)$ being a discontinuous solution satisfies the Rankine-Hugoniot shock condition [15]. The second possible solution is a continuous solution and is also known as a rarefaction wave type of solution. So that

$$v_2(x,t) = \begin{cases} 0 & if \;\; x \leq 0, \\ \dfrac{x}{t} & if \;\; 0 \leq x \leq t \\ 1 & if \;\; x \geq t \end{cases} \tag{2.19}$$

Here, as an example, we use the Lax's entropy condition [15, 56] to find the physically relevant weak solution of the inviscid Burgers' equation among two possible solutions, $v_1(x,t)$ and $v_2(x,t)$.

Theorem 2.1.1. *(Lax's entropy condition [15, 56]) The speed of the shock wave, which is calculated by using Rankine-Hugoniot shock conditions as explained in Chapter 1, satisfies the entropy condition if the following inequality*

$$f'(v_l) > \lambda > f'(v_r) \tag{2.20}$$

is satisfied. Here v_l and v_r are the constants on left and right sides of the discontinuity curve, respectively and λ is the speed of shock propagation.

Let's first consider whether the discontinuous solution $v_1(x,t)$ is an physically relevant weak solution. If discontinuous solution $v_1(x,t)$ satisfies both Rankine-Hugoniot shock condition and entropy condition, it is a physically relevant weak solution [56]. Since, $v_1(x,t)$ already satisfies Rankine-Hugoniot shock condition as mentioned above, it needs to be checked if it also satisfies the Lax's entropy condition in equation (2.20).

Thus, first, the speed of discontinuous propagation λ is calculated as follows

$$[f(v)] = \lambda[v], \tag{2.21}$$

where $[f(v)] = f(v_l) - f(v_r)$ and $[v] = v_l - v_r$. Then,

$$\frac{(v_l)^2 - (v_r)^2}{2} = \lambda(v_l - v_r) \tag{2.22}$$

$$\lambda = \frac{v_l + v_r}{2}. \tag{2.23}$$

After substituting v_l and v_r into equation (2.23),

$$\lambda = \frac{1}{2}.$$

Then, equation (2.20) becomes [59]

$$f'(v_l) = v_l = 0 \not> \frac{1}{2} \not> 1 = v_r = f'(v_r).$$

Consequently, it is clear that the Lax's entropy condition is violated by the solution $v_1(x,t)$ along the discontinuity curve, $x = t/2$. Therefore, the rarefaction wave solution, $v_2(x,t)$ can be selected as the physically relevant solution of the inviscid Burgers' equation.

Now, we consider a system with a convex entropy function $S(v(x,t))$ augmented with a corresponding entropy flux function $Q(v(x,t))_v = S_v^T g_v$, where $g_v = J(v)$ is the Jacobian matrix of the flux function $g(v(x,t))$ [60]. Multiplying equation (2.12) by the entropy variable $\mathbf{s}(v)^T = S_v$ gives the entropy conservation law

$$S_v v_t + S_v g(v)_x = 0,$$

or

$$S_t + Q_v v_x = 0 \tag{2.24}$$

$$S_t + Q_x = 0, \tag{2.25}$$

on using equation (2.12). Equation (2.25) is valid solely for smooth solutions. However, when flow encounters shock waves, a discontinuity appears in the system and the entropy should increase. To prevent this, first, the weak limiting solution of the modified conservation law is obtained by adding artificial viscosity to the right hand side (RHS) of equation (2.12) [61],

$$v^{\epsilon}(x,t)_t + g^{\epsilon}(v(x,t))_x = \epsilon v^{\epsilon}(x,t)_{xx}, \qquad x_L \leq x \leq x_R, \quad t \geq 0, \tag{2.26}$$

where $v^{\epsilon}(x,t)_{xx}$ is the artificial viscosity term. The limiting solution, i.e., $\lim_{\epsilon \to 0} v^{\epsilon}(x,t) = v(x,t)$, satisfies the original conservation law (equation (2.12)) weakly, when the constant ϵ of the artificial viscosity term goes to zero [60]. Second, an entropy inequality condition must be satisfied for the modified conservation law in order to obtain entropy stable solutions [60]. For this, the same procedure is applied as for obtaining the entropy conservative equation in the smooth regime. Equation (2.26) is multiplied by the entropy variable $\mathbf{s}(v)^T = S_v$, so that

$$S(v)_t + Q(v)_x = \epsilon S(v)_v v_{xx}. \tag{2.27}$$

The RHS of equation (2.27) can be rewritten as

$$S(v)_v v_{xx} = (S(v)_v v_x)_x - S(v)_{vv} v_x^2. \tag{2.28}$$

If the convexity property of the entropy function, $S(v)_{vv} \geq 0$, is applied in equation (2.28), this leads

$$(S(v)_v v_x)_x \geq S(v)_v v_{xx}. \tag{2.29}$$

After substituting equation (2.29) into equation (2.27)

$$S(v)_t + Q(v)_x \leq \epsilon (S(v)_v v_x)_x \tag{2.30}$$

and letting ϵ goes to zero, the entropy inequality condition is

$$S(v)_t + Q(v)_x \leq 0. \tag{2.31}$$

The main reason for making changes on the RHS of equation (2.27) is to prevent any uncontrolled discontinuity which might be created owing to the v_x^2 term after integrating equation (2.27) in space [15]. Then, the entropy inequality is also integrated in space to control the growth in physical entropy with time,

$$\frac{d}{dt}\int_{x_L}^{x_R} S(v) + Q(v(x_R,t)) - Q(v(x_L,t)) \leq 0. \tag{2.32}$$

The desired entropy stable solutions can be attained if any change of entropy is bounded in time. Thus, $Q(v(x_L,t))-Q(v(x_R,t)) \leq C$ has to be bounded above by a constant $C \in \mathbb{R}$ which depends on the initial values of the initial value problem. However, since initial-boundary value problems are the main subject in this study this constant depends on both the initial and boundary values of these problems. Therefore, only the left boundary at $x = x_L$ is considered in detail in the present chapter. The right boundary can be treated similarly. Hence, the entropy flux at the left boundary, $Q(v(x_L,t))$, must also be bounded.

Before applying this entropy theory in the numerical scheme for solving the hyperbolic conservation law, there are a couple of points to be highlighted. First, the conservation laws should be rewritten with entropy variables. This can be done by using the convexity property of entropy function, which leads to a one-to-one mapping between the conservative variable, v, and the entropy variable, $\mathbf{s}(v)$, i.e., $v \rightarrow \mathbf{s}(v)$ [26, 27]. Thus, conservation laws with conservative variables can be rewritten in terms of entropy variables as

$$v(\mathbf{s})_t + g(\mathbf{s})_x = 0, \tag{2.33}$$

where $v = v(\mathbf{s})$ and $g(v(\mathbf{s})) = g(\mathbf{s})$. Second, thanks to the convexity property of the entropy function, the governing system with entropy variables is symmetric hyperbolic [26, 27, 60]. This symmetric form of the conservation law with entropy variables has no negative effect on conservation and still preserves the same weak solution of the governing system [28]. Finally, another fundamental crucial function was reported by Harten [28]. Here, some Jacobian of nonlinear scalar functions with respect to entropy variables were given as the definition of the conservative variable, v, and the

corresponding conservative flux function, $g(v(\mathbf{s}))$, such that

$$\xi_{\mathbf{s}} = v^T, \qquad \eta_{\mathbf{s}} = g^T. \tag{2.34}$$

Here, T denotes the transpose and ξ and η are so-called *potential* and *potential flux* functions, respectively. These nonlinear functions are used in the description of the entropy and corresponding entropy flux functions [26, 28], such as

$$S(v) = v^T\mathbf{s} - \xi(\mathbf{s}) \tag{2.35}$$

$$Q(v) = g^T\mathbf{s} - \eta(\mathbf{s}). \tag{2.36}$$

An equivalent form was reported by [25, 27] as

$$\xi(\mathbf{s}) = \mathbf{s}^T v - S(v) \tag{2.37}$$

$$\eta(\mathbf{s}) = \mathbf{s}^T g - Q(v), \tag{2.38}$$

which also satisfy equation (2.34) [62], such that

$$\xi_{\mathbf{s}} = v^T + \mathbf{s}^T v_{\mathbf{s}} - S_v v_{\mathbf{s}} = v^T$$
$$\eta_{\mathbf{s}} = g^T + \mathbf{s}^T g_v v_{\mathbf{s}} - Q_v v_{\mathbf{s}} = g^T.$$

It is worth mentioning that these potential fluxes are also convex according to the convexity property of entropy variables (the inverse of the entropy variable is also convex), such that

$$\xi_{\mathbf{ss}} = v_{\mathbf{s}}^T = S_{vv}^{-1} \geq 0, \qquad v \to \mathbf{s}(v)^T = S_v,$$

by a one-to-one mapping between the entropy and conservative variables [25]. The potential function and corresponding potential flux function are particularly essential in determining whether a numerical scheme is entropy conservative or entropy stable. To achieve numerical entropy stable solutions, the associated entropy functions and the other auxiliary relations explained above are added to the numerical scheme in

the following sections. In Section 2.2 numerical entropy stable schemes are introduced briefly for an initial value problem (Cauchy case, $\Omega = (-\infty, +\infty)$), explained in the work of [25]. Then, in the following Section 2.3 the proposed scheme is derived by restricting the Cauchy domain with a left boundary, $\Omega = [x_L, +\infty)$, as described in the work of [29].

2.2 Overview of entropy stable schemes for Cauchy Case

In this section, the hyperbolic conservation law with entropy variables (equation 2.33) is semi-discretized by using the theory detailed in Section 2.1, so that

$$\frac{d}{dt}v(\mathbf{s}_i(t)) + \frac{g_{i+1/2} - g_{i-1/2}}{\Delta x} = 0, \tag{2.39}$$

where

$$g_{i+1/2} = \frac{g_{i+1} + g_i}{2} - \frac{D_{i+1/2}(\mathbf{s}_{i+1} - \mathbf{s}_i)}{2}. \tag{2.40}$$

Here, $\Delta x = x_{i+1} - x_i$ denotes the mesh size between two consecutive discrete point, and $g_{i+1/2}$ is the discrete numerical flux vector at $x_{i+1/2} = (x_i + x_{i+1})/2$. Moreover, $D_{i+1/2}$ is the numerical diffusion matrix and $v(\mathbf{s}_i)$ refers to the numerical solution vector at x_i. In addition, a consistency between the discrete (see equation (2.39)) and differential (see equation (2.33)) forms of the conservation laws is assumed to exist.

The numerical entropy stable solutions are required to satisfy the following numerical entropy stability condition locally and globally

$$(S_i)_t + \frac{Q_{i+1/2} - Q_{i-1/2}}{\Delta x} \leq 0. \tag{2.41}$$

To this end, equation (2.39) is multiplied by the discrete entropy variable $\mathbf{s}^T(v_i) = S_v(v_i)$ from the left, so that

$$\mathbf{s}_i^T(v_i)_t + \mathbf{s}_i^T \frac{g_{i+1/2} - g_{i-1/2}}{\Delta x} = 0. \tag{2.42}$$

After adding and subtracting $\frac{1}{2}\mathbf{s}_{i+1}^T g_{i+1/2}$ and $\frac{1}{2}\mathbf{s}_{i-1/2}^T g_{i-1/2}$ terms in equation (2.42),

we have

$$\Delta x(S_i)_t + \frac{1}{2}(\mathbf{s}_i + \mathbf{s}_{i+1})^T g_{i+1/2} + \frac{1}{2}(\mathbf{s}_i - \mathbf{s}_{i+1})^T g_{i+1/2} - \frac{1}{2}(\mathbf{s}_i + \mathbf{s}_{i-1})^T g_{i-1/2} - \frac{1}{2}(\mathbf{s}_i - \mathbf{s}_{i-1})^T g_{i-1/2} = 0. \tag{2.43}$$

Then, the scheme has the following numerical entropy flux function, as introduced in [25],

$$Q_{i+1/2} = \frac{1}{2}(\mathbf{s}_i + \mathbf{s}_{i+1})^T g_{i+1/2} - \frac{1}{2}(\eta_{i+1} + \eta_i). \tag{2.44}$$

This is also similar to the form (2.38). After substituting equation (2.44) into equation (2.43), we have

$$\Delta x(S_i)_t + Q_{i+1/2} + \frac{1}{2}(\mathbf{s}_i - \mathbf{s}_{i+1})^T g_{i+1/2} + \frac{1}{2}(\eta_{i+1} + \eta_i) - Q_{i-1/2} - \frac{1}{2}(\mathbf{s}_i - \mathbf{s}_{i-1})^T g_{i-1/2} - \frac{1}{2}(\eta_i + \eta_{i-1}) = 0, \tag{2.45}$$

or, equivalently,

$$\Delta x(S_i)_t + Q_{i+1/2} + \frac{1}{2}(\mathbf{s}_i - \mathbf{s}_{i+1})^T g_{i+1/2} + \frac{1}{2}(\eta_{i+1} - \eta_i) - Q_{i-1/2} - \frac{1}{2}(\mathbf{s}_i - \mathbf{s}_{i-1})^T g_{i-1/2} + \frac{1}{2}(\eta_i - \eta_{i-1}) = 0. \tag{2.46}$$

Here, there is a need to state the comparison principle theorem presented in [25] for numerical fluxes which are entropy conservative or entropy stable. By using this theorem the amount of numerical viscosity possessed by the entropy conservative and entropy stable numerical fluxes can be compared.

Theorem 2.2.1. *The conservative scheme equation (2.39) is entropy conservative if, for three-point schemes,*

$$\Delta \boldsymbol{s}_{i+1/2}^T g_{i+1/2} = \Delta \eta_{1+1/2}, \tag{2.47}$$

and entropy stable if the inequality

$$\Delta \boldsymbol{s}_{i+1/2}^T g_{i+1/2} \leq \Delta \eta_{1+1/2} \tag{2.48}$$

holds, where $\Delta \boldsymbol{s}_{i+1/2}^T = (\boldsymbol{s}_{i+1} - \boldsymbol{s}_i)^T/2$ *and* $\Delta\eta_{i+1/2} = (\eta_{i+1} - \eta_i)/2$ *(see [25]).*

Here, the three-point scheme refers to the numerical solution, v_i^{n+1} which depends on the three numerical values from the previous time step, v_{i-1}^n, v_i^n and v_{i+1}^n. According to the study in [25] if the amount of viscosity is greater than the one possessed by numerical entropy conservative flux (2.47), then the numerical flux is so-called entropy stable, so that the condition in equation (2.48) is satisfied. Therefore, the comparison of viscosity can be stated as in [25]

$$D_{i+1/2} \leq D_{i+1/2}^{es}, \tag{2.49}$$

where $D_{i+1/2}$ and $D_{i+1/2}^{es}$ refer to viscosity in entropy conservative and entropy stable fluxes for the conservative numerical scheme (2.39), respectively. Letting $g_{i+1/2}$ be entropy stable and applying it in equation (2.46), the local entropy stability condition is

$$(S_i)_t + \frac{Q_{i+1/2} - Q_{i-1/2}}{\Delta x} \leq 0. \tag{2.50}$$

The sum of all locally entropy stable inequalities gives the global entropy inequality. Therefore, summation by parts and integrated in time are used to derive the global entropy stability condition

$$\sum_{i \in N} \Delta x S(v_i(t)) \leq \sum_{i \in N} \Delta x S(v_i(0)). \tag{2.51}$$

In this section, since initial value problems are considered, the global entropy bound is satisfied automatically with compactly supported initial data in RHS of the above equation. There are many examples of this given in [25, 38, 39].

2.3 Entropy stable schemes for a bounded domain

In this chapter, the main intention is to derive a stable numerical scheme having a global bound on the entropy. To this end, the entire scheme needs to satisfy the entropy stability condition in equation (2.41). To this point, the interior scheme is merely evaluated to satisfy the numerical entropy stability condition locally. Since the governing system is an initial boundary value PDE, the remaining boundary discretization must

satisfy the entropy stability condition and must lead to a global entropy bound. For simplicity, the boundary scheme is only considered at the left boundary, $x = x_L$ since the same procedure can be applied to the right boundary, $x = x_R$. Therefore, the numerical boundary scheme for the left boundary is

$$(v(\mathbf{s}_L))_t + \frac{g_{1/2} - \tilde{g}_L}{\Delta x/2} = 0. \tag{2.52}$$

Here, $v(\mathbf{s}_L)$ is the numerical solution and $\tilde{g}_L$ is the numerical left boundary function at $x = x_L$. Before giving a detailed prescription of the numerical boundary flux $\tilde{g}_L$, there is a need to define some auxiliary relations. Then the entropy stability condition needs to be satisfied for the boundary scheme (2.52). Hence, equation (2.52) is multiplied by $\mathbf{s}_L^T$, so that

$$\mathbf{s}_L^T(v(\mathbf{s}_L))_t + \mathbf{s}_L^T\frac{g_{1/2} - \tilde{g}_L}{\Delta x/2} = 0, \tag{2.53}$$

or

$$\frac{\Delta x}{2}\mathbf{s}_L^T(v(\mathbf{s}_L))_t + \mathbf{s}_L^T g_{1/2} - \mathbf{s}_L^T\tilde{g}_L = 0. \tag{2.54}$$

To satisfy the entropy stability criteria, $\frac{1}{2}v_1^T$ and $\frac{1}{2}(\eta_L+\eta_1)$ are added to and subtracted from the above equation, so that

$$\frac{\Delta x}{2}(S_L)_t + Q_{1/2} + \frac{1}{2}(\mathbf{s}_L - \mathbf{s}_1)^T g_{1/2} + \frac{1}{2}(\eta_1 + \eta_L) - \mathbf{s}_L^T\tilde{g}_L = 0. \tag{2.55}$$

Here, the numerical entropy flux function at the left boundary, $\tilde{Q}_L$, is determined as

$$\tilde{Q}_L = \mathbf{s}_L^T\tilde{g}_L - \eta_L, \tag{2.56}$$

where η_L is the potential flux function at the left boundary. Equation (2.55) then becomes

$$\frac{\Delta x}{2}(S_L)_t + Q_{1/2} - \tilde{Q}_L + \frac{1}{2}(\mathbf{s}_L - \mathbf{s}_1)^T g_{1/2} + \frac{1}{2}(\eta_1 - \eta_L) = 0.$$

It is assumed that the interior numerical fluxes ($g_{i+1/2}$ when $i \geq 0$) are entropy stable and so satisfy equation (2.48). The desired local entropy inequality is then obtained

for the left boundary as

$$(S_L)_t + \frac{Q_{1/2} - \tilde{Q}_L}{\Delta x/2} \leq 0. \tag{2.57}$$

Next, we multiply the inner (2.50) and left boundary (2.57) entropy schemes by Δx and $\Delta x/2$ respectively and sum to obtain,

$$\frac{\Delta x}{2}(S_L)_t + \sum_{i>0} \Delta x (S_i)_t \leq \tilde{Q}_L = \mathbf{s}_L^T \tilde{g}_L - \eta_L. \tag{2.58}$$

The numerical boundary entropy flux function at the RHS has to be bounded to provide a global entropy bound, $\tilde{Q}_L < C$, where $C \in \mathbb{R}$ is a constant. Therefore, the entropy stable numerical boundary flux, $\tilde{g}_L$, has to be constructed to bound $\tilde{Q}_L$. Theorem 2.2.1 is presented as a recipe for constructing entropy stable fluxes using the comparison principle for numerical diffusion matrices. Therefore, first the numerical entropy conservative boundary flux is described and then the entropy stable flux due to adding artificial diffusion into the system is explained.

Let $\hat{g}_L$ be an entropy conservative boundary flux which satisfies

$$(\mathbf{s}_L - \mathbf{s}(\mathbf{b}_L))^T \hat{g}_L = \eta_L - \eta(\mathbf{b}_L). \tag{2.59}$$

This step alone is not enough for the numerical solution to converge to physical weak solutions, since there are many choices of entropy conservative functions, even only one entropy is used [25, 39]. The entropy conservative fluxes are also not enough to enforce a decay in the entropy across a discontinuity. In [63], the Roe averaged matrix was implemented into the boundary flux as an artificial numerical diffusion to construct an entropy stable boundary flux. The original Roe matrix was constructed for conservative fluxes to satisfy

$$g(v_{i+1}) - g(v_i) = \tilde{R}_{i+1/2}(v_{i+1} - v_i), \tag{2.60}$$

where $g(v_i)$ is the numerical flux in conservative variables. The advantage of using the Roe average matrix is that it enables the capture of a shock wave satisfying the Rankine-Hugoniot jump condition [63]. This condition is an example of how two grid

points are connected across a shock or a contact discontinuity. In general, it is

$$g(v_{i+1}) - g(v_i) = V(v_{i+1} - v_i), \tag{2.61}$$

where V is the velocity of the shock wave [50]. The Roe average matrix also preserves conservation and does not have a complicated algebra structure [63].

In [25, 62] the existence of such a matrix is given in the general case by integrating the Jacobian matrix over a path between v_i and v_{i+1}, i.e., $v_{i+1/2} = \frac{1}{2}(v_i + v_{i+1}) + \theta\Delta v_{i+1/2}$, where $\Delta v_{i+1/2} = (v_{i+1} - v_i)$. Hence,

$$g(v_{i+1}) - g(v_i) = \int_{-1/2}^{1/2} g_v(v_{i+1/2}(\theta))v_\theta(\theta)d\theta, \tag{2.62}$$

where $v_\theta = \Delta v_{i+1/2}$ refers to derivative of v with respect to θ and is independent of θ. After substituting equation (2.62) into equation (2.60), the integral form of the Jacobian matrix can be written in general form as

$$\tilde{R}_{i+1/2} = \int_{-1/2}^{1/2} g_v(v_{i+1/2}(\theta))d\theta. \tag{2.63}$$

Since the Roe average matrix is introduced into the numerical scheme as a numerical artificial diffusion matrix with conservative variables, the mapping between the conservative and entropy variables is defined via an auxiliary integral along the same path, $\mathbf{s}_{i+1/2} = \frac{1}{2}(\mathbf{s}_i + \mathbf{s}_{i+1}) + \theta\Delta\mathbf{s}_{i+1/2}$. Therefore,

$$v_{i+1} - v_i = \int_{-1/2}^{1/2} v(\mathbf{s}_{i+1/2}(\theta))_\theta d\theta = \int_{-1/2}^{1/2} v_{\mathbf{s}}(\mathbf{s}_{i+1/2}(\theta))d\theta(\mathbf{s}_{i+1} - \mathbf{s}_i), \tag{2.64}$$

where $P_{i+1/2} = \int_{-1/2}^{1/2} v_{\mathbf{s}}(\mathbf{s}_{i+1/2}(\theta))d\theta$ is symmetric invertible and positive definite matrix and has a one-to-one connection between the conservative and entropy variables due to the convexity of the physical entropy [25]. Then, substituting equations (2.63) and (2.64) into equation (2.60), we have

$$\begin{aligned} g(v_{i+1}) - g(v_i) &= \tilde{R}_{i+1/2}(v_{i+1} - v_i) \\ &= \int_{-1/2}^{1/2} g_v(v_{i+1/2}(\theta))d\theta \int_{-1/2}^{1/2} v_{\mathbf{s}}(\mathbf{s}_{i+1/2}(\theta))d\theta(\mathbf{s}_{i+1} - \mathbf{s}_i) \end{aligned}$$

$$= \int_{-1/2}^{1/2} g_{\mathbf{s}}(\mathbf{s}_{i+1/2}(\theta)) d\theta (\mathbf{s}_{i+1} - \mathbf{s}_i)$$
$$= \tilde{G}_{i+1/2}(\mathbf{s}_{i+1} - \mathbf{s}_i),$$

where $\tilde{G}_{i+1/2} = \int_{-1/2}^{1/2} g_{\mathbf{s}}(\mathbf{s}_{i+1/2}(\theta)) d\theta$ is the Jacobian matrix for the flux in entropy variables $g_{\mathbf{s}}$. Then, the connection of the Roe average Jacobian fluxes with the entropy and conservative variables is

$$\tilde{R}_{i+1/2} = \tilde{G}_{i+1/2} P^{-1}_{i+1/2}. \tag{2.65}$$

Since $\tilde{G}_{i+1/2}$ is a symmetric Roe average matrix, it is diagonalizable as

$$\tilde{G}_{i+1/2} = \tilde{K}_{i+1/2} \tilde{\tau}_{i+1/2} \tilde{K}^T_{i+1/2}, \tag{2.66}$$

where $\tilde{K}^T_{i+1/2}$ and $\tilde{K}_{i+1/2}$ are normalized eigenvectors, $\tilde{K}^T_{i+1/2} \tilde{K}_{i+1/2} = I$. Here, I is the identity matrix. It is to be remarked that the Roe average matrix between the boundary value and boundary data has $g(\mathbf{s}_L) - g(\mathbf{b}_L) = \tilde{G}_L(\mathbf{s}_L - \mathbf{b}_L)$.

The numerical entropy conservative boundary flux for the left boundary is constructed as

$$\hat{g}_L = \frac{g(\mathbf{s}_L) + g(\mathbf{b}_L)}{2} - \frac{\hat{D}_L}{2}(\mathbf{s}_L - \mathbf{b}_L), \tag{2.67}$$

where $\hat{D}_L$ is an unknown symmetric matrix. In the case of equality between $\mathbf{s}_L$ and $\mathbf{b}_L$ the numerical entropy conservative boundary flux $\hat{g}_L$ can be written as an arithmetic average of two fluxes of boundary value and boundary data, $\frac{g(\mathbf{s}_L)+g(\mathbf{b}_L)}{2}$. The matrix $\hat{D}_L$ is assumed to have the same eigenvectors as the Roe average matrix, i.e., $\hat{D}_L = \tilde{K}_L \hat{\tau}_L \tilde{K}^T_L$, where $\hat{\tau}_L$ is an unknown diagonal matrix [29]. Since the eigenvectors of the Roe matrix are known, the remaining terms to be determined are the eigenvalues of the matrix $\hat{D}_L$. Hence, the definition of the $\hat{D}_L$ matrix is inserted into equation (2.67), to yield

$$\tilde{K}^T_L \left(\hat{g}_L - \frac{g(\mathbf{s}_L) + g(\mathbf{b}_L)}{2} \right) = -\frac{\hat{\tau}_L}{2} \tilde{K}^T_L (\mathbf{s}_L - \mathbf{b}_L), \tag{2.68}$$

where $\hat{\tau}_L$ has diagonal entries which are the positive or negative diffusion coefficients of the entropy conservative scheme. It is remarked that, in general, entropy conservative

flux has no diffusion and the $\hat{D}_L$ matrix is a resemblance of the diffusion $D_{i+1/2}$ at the boundary scheme definition in equation (2.40). This is clarified by considering the example of the inviscid Burgers equation in Appendix A. The numerical left boundary flux is derived by considering only in going characteristics. Therefore, only positive eigenvalues are considered and the left numerical boundary flux is introduced explicitly as in [29],

$$\tilde{g}_L = \frac{g(\mathbf{s}_L) + g(\mathbf{b}_L)}{2} - \frac{Y_L}{2}(\mathbf{s}_L - \mathbf{b}_L), \tag{2.69}$$

where $Y_L = |\tilde{G}_L| + \hat{D}_L + |\hat{D}_L|$ and $|\tilde{G}_L|$ have all eigenvalues in a positive sign, i.e., $|\tilde{G}_L| = \tilde{K}_L|\tilde{\tau}_L|\tilde{K}_L^T$. Also, positive eigenvalues of the diffusion matrix of the entropy conservative flux are used with $\hat{D}^+ = \frac{\hat{D}_L + |\hat{D}_L|}{2} = \tilde{K}_L\hat{\tau}_L^+\tilde{K}_L^T$, where $\hat{\tau}_L^+$ represents the positive eigenvalues.

The stability of the solution of non-linear problems is proved by capturing any possible growth of the entropy. In another words, by having a global bound on entropy. This is achieved by having a bound on the boundary terms for initial boundary value problems. Therefore, the following theorem is an essential guide for achieving the desired stability for numerical solutions of the governing conservation law.

Theorem 2.3.1. *[29] Let the numerical scheme for the governing initial-boundary value problem (2.12)-(2.14) be consistent numerical schemes for the interior and boundary parts defined as*

$$v(\boldsymbol{s}_i)_t + \frac{g_{i+1/2} - g_{i-1/2}}{\Delta x} = 0, \qquad i > 0 \tag{2.70}$$

and

$$v(\boldsymbol{s}_L)_t + \frac{g_{1/2} - \tilde{g}_L}{\Delta x/2} = 0. \tag{2.71}$$

Let $g_{i+1/2}$, $i \geq 0$, be consistent entropy stable fluxes and the entropy stability conditions (2.50) and (2.57) hold for the interior and boundary schemes, respectively. Then, the desired global bound is obtained if the following inequality is satisfied

$$\frac{\Delta x}{2}(S_L)_t + \Delta x \sum_{i>0}(S_i)_t \leq \tilde{Q}_L = \boldsymbol{s}_L^T\tilde{g}_L - \eta_L \leq C. \tag{2.72}$$

An L^2 bound on the solution, $\|v\|_2 \leq C_1$ [31], where $C, C_1 \in \mathbb{R}$ are constants then

follows [29].

Proof. It is clear that the RHS of equation (2.72) has to be bounded to obtain a global entropy bound. A similar argument with a different entropy function, called homogeneous entropy, is used for the compressible Euler equations in [35]. The reason is that there are several options for entropy functions to symmetrize the Euler equations of gas dynamics, whilst there is solely one entropy function (physical entropy function) to symmetrize the Navier-Stokes equations. Since the present work is considered as a steeping stone towards obtaining numerical results for the Navier-Stokes equations, the proof of this theorem is different than the proof of [35]. The present Euler equations can be treated as the convective part of the Navier-Stokes equations.

Let us start by adding and subtracting $Q(\mathbf{b}_L)$ to the definition (2.56) of the numerical boundary entropy flux function

$$\tilde{Q}_L = \mathbf{s}_L^T \tilde{g}_L - \eta_L + Q(\mathbf{b}_L) - Q(\mathbf{b}_L),$$

where $Q(\mathbf{b}_L) = \mathbf{s}(\mathbf{b}_L)^T g(\mathbf{b}_L) - \eta(\mathbf{b}_L)$. Then,

$$\begin{aligned} \tilde{Q}_L &= \mathbf{s}_L^T \tilde{g}_L - \eta_L - \mathbf{s}(\mathbf{b}_L)^T g(\mathbf{b}) + \eta(\mathbf{b}_L) + Q(\mathbf{b}_L) \\ &= (\mathbf{s}_L - \mathbf{s}(\mathbf{b}_L))^T \tilde{g}_L + \mathbf{s}(\mathbf{b}_L)^T (\tilde{g}_L - g(\mathbf{b}_L)) - (\eta_L - \eta(\mathbf{b}_L)) + Q(\mathbf{b}_L). \end{aligned}$$

After using the definition of numerical boundary flux from equation (2.69), this becomes

$$\begin{aligned} \tilde{Q}_L = \mathbf{s}(\mathbf{b}_L)^T (\tilde{g}_L - g(\mathbf{b}_L)) + Q(\mathbf{b}_L) - (\eta_L - \eta(\mathbf{b}_L)) \\ + (\mathbf{s}_L - \mathbf{s}(\mathbf{b}_L))^T \left(\hat{g}_L - \frac{|\tilde{G}_L| + |\hat{D}_L|}{2} (\mathbf{s}_L - \mathbf{s}(\mathbf{b}_L)) \right), \end{aligned} \tag{2.73}$$

where the numerical entropy conservative boundary flux satisfies

$$\begin{aligned} &(\mathbf{s}_L - \mathbf{s}(\mathbf{b}_L))^T \hat{g}_L = (\eta_L - \eta(\mathbf{b}_L)), \\ &\quad = \mathbf{s}(\mathbf{b}_L)^T (\tilde{g}_L - g(\mathbf{b}_L)) + Q(\mathbf{b}_L) - (\mathbf{s}_L - \mathbf{s}(\mathbf{b}_L))^T \frac{|\tilde{G}_L| + |\hat{D}_L|}{2} (\mathbf{s}_L - \mathbf{s}(\mathbf{b}_L)). \end{aligned}$$

Here, $(\tilde{g}_L - g(\mathbf{b}_L))$ is calculated from

$$\tilde{g}_L = \frac{g(\mathbf{s}_L) + g(\mathbf{b}_L)}{2} - \frac{|\tilde{G}_L| + \hat{D}_L + |\hat{D}_L|}{2}(\mathbf{s}_L - \mathbf{b}_L)$$
$$= g(\mathbf{b}_L) - \frac{g(\mathbf{b}_L) - g(\mathbf{s}_L)}{2} - \frac{|\tilde{G}_L| + \hat{D}_L + |\hat{D}_L|}{2}(\mathbf{s}_L - \mathbf{b}_L),$$

where $g(\mathbf{s}_L) - g(\mathbf{b}_L) = \tilde{G}_L(\mathbf{s}_L - \mathbf{b}_L)$ by the definition of the Roe average matrix between the boundary point and boundary data. Hence

$$\tilde{g}_L = g(\mathbf{b}_L) + \frac{\tilde{G}_L}{2}(\mathbf{s}_L - \mathbf{b}_L) - \left(\frac{|\tilde{G}_L|}{2} + \hat{D}_L^+\right)(\mathbf{s}_L - \mathbf{b}_L)$$
$$= g(\mathbf{b}_L) + (\tilde{G}_L^- - \hat{D}_L^+)(\mathbf{s}_L - \mathbf{b}_L).$$

Inserting the above expression into (2.73) gives,

$$\tilde{Q}_L = \mathbf{s}(\mathbf{b}_L)^T(\tilde{G}_L^- - \hat{D}_L^+)(\mathbf{s}_L - \mathbf{s}(\mathbf{b}_L)) + Q(\mathbf{b}_L) - (\mathbf{s}_L - \mathbf{s}(\mathbf{b}_L))^T\frac{|\tilde{G}_L| + |\hat{D}_L|}{2}(\mathbf{s}_L - \mathbf{s}(\mathbf{b}_L)). \tag{2.74}$$

The growth of the solution affects the eigenvalues as they depend on the solution of the non-linear PDE [35, 41]. Therefore, equation (2.74) is rewritten using eigencomponents, i.e., eigenvalue and eigenvectors, of the $\tilde{G}_L$ and $\hat{D}_L$ matrices,

$$\tilde{Q}_L = Q(\mathbf{b}_L) + \mathbf{s}(\mathbf{b}_L)^T(\tilde{K}_L(\tilde{\tau}_L^- - \hat{\tau}_L^+)\tilde{K}_L^T)(\mathbf{s}_L - \mathbf{s}(\mathbf{b}_L))$$
$$- (\mathbf{s}_L - \mathbf{s}(\mathbf{b}_L))^T\frac{\tilde{K}_L(|\tilde{\tau}_L| + |\hat{\tau}_L|)\tilde{K}_L^T}{2}(\mathbf{s}_L - \mathbf{s}(\mathbf{b}_L)). \tag{2.75}$$

For notational convenience the subscript L is now dropped. The eigenvalue and eigenvector matrices have N components such that $\hat{\tau} = \text{diag}(\hat{\tau}_1, \ldots, \hat{\tau}_n)$, $\tilde{\tau} = \text{diag}(\tilde{\tau}_1, \ldots, \tilde{\tau}_n)$ and $\tilde{K} = \left[\tilde{k}_1 | \ldots | \tilde{k}_n\right]$, where $\hat{\tau}_i$, $\tilde{\tau}_i$ and $\tilde{k}_i$ represent ith eigenvalues and eigenvector, respectively. We define $\mathbf{s}(\mathbf{b})_i = \mathbf{s}(\mathbf{b})^T\tilde{k}_i$ and $(\mathbf{s} - \mathbf{s}(\mathbf{b}))_i = (\mathbf{s} - \mathbf{s}(\mathbf{b}))^T\tilde{k}_i$ and insert these into equation (2.75), so that

$$\tilde{Q}_L = Q(\mathbf{b}) + \sum_{i=1}^{N}\mathbf{s}(\mathbf{b})_i(\tilde{\tau}_i^- - \hat{\tau}_i^+)(\mathbf{s} - \mathbf{s}(\mathbf{b}))_i^T - \frac{1}{2}(\mathbf{s} - \mathbf{s}(\mathbf{b}))_i(|\tilde{\tau}_i| + |\hat{\tau}_i|)(\mathbf{s} - \mathbf{s}(\mathbf{b}))_i^T. \tag{2.76}$$

It can be observed that the last term in equation (2.76) is non-positive and $\tilde{\tau}_i^- - \hat{\tau}_i^+ \leq |\tilde{\tau}_i| + |\hat{\tau}_i|$, where $|\tilde{\tau}_i| + |\hat{\tau}_i| \geq 0$. However, this result is not adequate to fulfill the requirements to satisfy an upper bound on entropy $\tilde{Q}_L$. All possible growth rates of

either eigenpairs or solution values have to be evaluated. It is noted that the argument is now developed by evaluating each eigenvalue contribution separately for simplicity, since both can be dealt with similarly. Hence

$$\tilde{Q}_L = Q(\mathbf{b}) + BT_{\tilde{\tau}} + BT_{\hat{\tau}},$$

where

$$BT_{\hat{\tau}} = \sum_{i=1}^{N} -\mathbf{s}(\mathbf{b})_i(\hat{\tau}_i^+)(\mathbf{s}-\mathbf{s}(\mathbf{b}))_i^T - \frac{1}{2}(\mathbf{s}-\mathbf{s}(\mathbf{b}))_i(|\hat{\tau}_i|)(\mathbf{s}-\mathbf{s}(\mathbf{b}))_i^T \tag{2.77}$$

and

$$BT_{\tilde{\tau}} = \sum_{i=1}^{N} \mathbf{s}(\mathbf{b})_i(\tilde{\tau}_i^-)(\mathbf{s}-\mathbf{s}(\mathbf{b}))_i^T - \frac{1}{2}(\mathbf{s}-\mathbf{s}(\mathbf{b}))_i(|\tilde{\tau}_i|)(\mathbf{s}-\mathbf{s}(\mathbf{b}))_i^T \tag{2.78}$$

are boundary terms corresponding to the $\hat{\tau}$ and $\tilde{\tau}$ eigenvalues, respectively. Here, only $BT_{\tilde{\tau}}$ is considered. Further details on $BT_{\hat{\tau}}$ can be found in [29].

$BT_{\tilde{\tau}}$ is decomposed into two groups of boundary terms for the contribution of the $\tilde{\tau}$ eigenvalues, since each direction is independent [63]. They are denoted by

$$BT_1 = \mathbf{s}(\mathbf{b})_1(\tilde{\tau}_1^-)(\mathbf{s}-\mathbf{s}(\mathbf{b}))_1^T - \frac{1}{2}(\mathbf{s}-\mathbf{s}(\mathbf{b}))_1(|\tilde{\tau}_1|)(\mathbf{s}-\mathbf{s}(\mathbf{b}))_1^T \tag{2.79}$$

$$BT_{2N} = \sum_{i=2}^{N} \mathbf{s}(\mathbf{b})_i(\tilde{\tau}_i^-)(\mathbf{s}-\mathbf{s}(\mathbf{b}))_i^T - \frac{1}{2}(\mathbf{s}-\mathbf{s}(\mathbf{b}))_i(|\tilde{\tau}_i|)(\mathbf{s}-\mathbf{s}(\mathbf{b}))_i^T. \tag{2.80}$$

Without loss of generality, BT_{2N} is assumed to be bounded via bounded eigenvalues $\tilde{\tau}_i$, $i = 2, .., N$. BT_1 is considered below in the rest of the proof. This is a convenient and simple way to discuss the existence of any potential growth in the system, because of the fact that each eigenvalue direction is evaluated individually. If there is a similar behavior in one of the eigenvalue directions of the BT_{2N}, this can be treated similarly as for the BT_1. Next, two main cases are considered for the BT_1 boundary term.

Case 1:

Assume that the eigenvalue $\tilde{\tau}_1$ is bounded. Then, the remaining solution values $\mathbf{s}(\mathbf{b})_1$ and $(\mathbf{s}-\mathbf{s}(\mathbf{b}))_1$ are evaluated to determine the possibility of any possible growth. Here, recall that $\mathbf{s}(\mathbf{b})_1$ is introduced via the bounded physically admissible boundary data, $\mathbf{b} \in L^\infty$ and $\tilde{k}_1$ is a unit vector. Therefore, $\mathbf{s}(\mathbf{b})_1$ is bounded. Then, the remaining

$(\mathbf{s}-\mathbf{s}(\mathbf{b}))_1$ solution value is considered in two cases as well. If $(\mathbf{s}-\mathbf{s}(\mathbf{b}))_1$ is bounded, then there are constants which bound the growth from above and below for the BT_1 boundary terms such that

$$C_1 \leq BT_1 \leq C,$$

where $C \in \mathbb{R}$ and $C_1 \in \mathbb{R}$ are the constants. It is worth noting that it is sufficient to obtain a bound from only above on $\tilde{Q}_L$ to obtain an entropy stable solution. Also, if $(\mathbf{s}-\mathbf{s}(\mathbf{b}))_1$ grows rapidly enough, then the last term on the RHS of BT_1 in equation (2.79) results in a negative quadratic growth. Hence, BT_1 is bounded from above with

$$BT_1 \leq C.$$

Case 2:

Let the $|\tilde{\tau}_1|$ eigenvalue has unbounded growth, i.e., $|\tilde{\tau}_1| \to \infty$. Then similar arguments to Case 1 can be applied. In terms of the growth of $|(\mathbf{s}-\mathbf{s}(\mathbf{b}))_1| \geq 2|\mathbf{s}(\mathbf{b})_1|$, the quadratic terms suppress the remaining terms and give an upper bound on BT_1 independently of $|\tilde{\tau}_1|$, so that

$$BT_1 \leq C.$$

However, if $|(\mathbf{s}-\mathbf{s}(\mathbf{b}))_1|$ is bounded above where $2|\mathbf{s}(\mathbf{b})_1|$, i.e., $|(\mathbf{s}-\mathbf{s}(\mathbf{b}))_1| \leq 2|\mathbf{s}(\mathbf{b})_1|$, we have that $BT_1 > 0$ can grow rapidly. However, this is theoretically not possible for conservation laws which are subject to entropy. This is because that any growth of the eigenvalues $\tilde{\tau}_1$ will cause a growth in its eigenvector, $\tilde{k}_1$ and also its solution $(\mathbf{s}-\mathbf{s}(\mathbf{b}))_1 = (\mathbf{s}-\mathbf{s}(\mathbf{b}))^T\tilde{k}_1$. This can be proved by expressing the governing conservation law as

$$v_{\mathbf{s}}\mathbf{s}_t + g_{\mathbf{s}}\mathbf{s}_x = 0 \tag{2.81}$$

so that,

$$\begin{aligned} v_{\mathbf{s}}\mathbf{s}_t + \tilde{K}\tilde{\tau}\tilde{K}^T\mathbf{s}_x &= 0 && (2.82)\\ \tilde{K}^T v_{\mathbf{s}}\tilde{K}(\tilde{K}^T\mathbf{s})_t + \tilde{\tau}(\tilde{K}^T\mathbf{s})_x &= 0. && (2.83) \end{aligned}$$

where $v_{\mathbf{s}} > 0$ is a positive definite, symmetric matrix and $g_{\mathbf{s}}$ is a symmetric matrix due

to the convex physical entropy [25]. If any growth can occur, then it can only happen locally [29]. For this reason, the coefficients of the solution are frozen such that

$$\bar{K}^T \bar{v_{\mathbf{s}}} \bar{K} (\bar{K}^T \mathbf{s})_t + \bar{\tau} (\bar{K}^T \mathbf{s})_x = 0, \tag{2.84}$$

where all terms with the $^-$ superscript are frozen coefficients. Since $\bar{K}^T \bar{v_{\mathbf{s}}} \bar{K} > 0$, then the eigencomponents and solution grow positively. Therefore, $(\mathbf{s}-\mathbf{s}(\mathbf{b}))_1 = (\mathbf{s}-\mathbf{s}(\mathbf{b}))^T \tilde{k}_1$ causes a growth due to growth of the corresponding eigenvalue $|\tilde{\tau}_1|$ and so gives a bound on the entropy.

In conclusion, any possibility of growth on boundary terms according to the contributions of the $\tilde{\tau}$ and $\hat{\tau}$ eigenvalues has been examined and it was found that the numerical boundary entropy function is bounded above, $\tilde{Q}_L < C$. This leads to a global bound on the entropy and L^2 stability of the solution $\|v\|_2$ will follow according to the convex physical entropy detailed in [31]. □

Next, the following proposition is a brief description showing how boundary conditions are included in the scheme by using a consistent linear theory.

Proposition 2.3.1. *The proposed left boundary condition $\tilde{g}_L$ is consistent with linear theory and enforces the boundary data $\mathbf{b}_L$ on the characteristic variables for in-going fluxes [29].*

Proof. Linear theory indicates that if the solution is smooth, then linear stability can be implied for the non-linear system [64]. Therefore, linearization of the scheme is carried out to show how the boundary conditions are enforced around smooth solutions. Interior and boundary fluxes are written in a linear form as

$$g_{i+1/2} = \frac{g_i^{\mathbb{L}} + g_{i+1}^{\mathbb{L}}}{2} - \frac{D_{i+1/2}^{\mathbb{L}}}{2}(\mathbf{z}_{i+1} - \mathbf{z}_i) \tag{2.85}$$

$$\tilde{g}_L = A(x_L, t)\mathbf{z}_L - A^+(x_L, t)(\mathbf{z}_L - \mathbf{z}(\mathbf{b}_L)), \tag{2.86}$$

where A is a bounded matrix and $\mathbf{z}$ is a linear variable. In the linear case flux is defined by $g^{\mathbb{L}} = A(x,t)\mathbf{z}$ and the linear diffusion matrix is determined by $D^{\mathbb{L}}$. Here, the diffusion matrix at the boundary is considered as $D_{x=x_L}^{\mathbb{L}} = |A(x_L, t)|$ and $\mathbf{z}(\mathbf{b}_L)$ is the boundary data in the linearized scheme. In equation (2.86) boundary data is

included by in-going characteristics variables in A^+. This directly enforces the correct number of boundary conditions [29]. Moreover, it is noted that the boundary scheme for the non-linear case in equation (2.52) is changed to

$$(v_L)_t + \frac{A(x_1,t)\mathbf{z}_1 - A(x_L,t)\mathbf{z}_L}{\Delta x/2} + \frac{A^+(x_L,t)}{\Delta x/2}(\mathbf{z}_L - \mathbf{z}(\mathbf{b}_L)) = 0. \tag{2.87}$$

Then, this scheme reduces to the standard SBP-SAT schemes that have been extensively analyzed for the linearized Euler and Navier-Stokes equations [65, 66]. □

Lastly, a convergence weak solution is obtained with the conservative numerical scheme by using the theory developed above. The proof is presented in the next section.

2.3.1 Lax Wendroff Theorem

The numerical solution converges to a physically relevant weak solution as the mesh size tends to zero, i.e., $\Delta x, \Delta t \to 0$ for the conservative entropy stable numerical scheme. To do this a well known Lax-Wendroff Theorem [15, 29, 34] is proved for the governing equations, which are non-linear hyperbolic conservation laws in a bounded domain. It is worth noting that the existence of physically relevant weak solutions comes from the numerical scheme satisfying the entropy stability criteria.

Theorem 2.3.2. *A sequence of bounded numerical solutions $v_i(x,t)$, i.e., $v_i(x,t) \in L^\infty$, is generated by a consistent and conservative entropy stable numerical method as given by equations (2.39), (2.52) and (2.69). Then, this sequence of numerical solutions $v_i(x,t)$ converges almost everywhere to a weak solution as given by equation (2.15) as the grid is refined, i.e., $\Delta x, \Delta t \to 0$, including the initial and boundary values.*

Proof. Let $\phi(x,t)$ be a compactly supported smooth test function, i.e., $\phi(x,t) \in C_0^1$ which is projected onto discrete points, $\phi_i(x_i,t) = \phi(x_i,t)$. Then, the interior and boundary parts of the numerical scheme given by equations (2.39) and (2.52) are multiplied by $\phi_i \Delta x$ and $\phi_0 \frac{\Delta x}{2}$, respectively, to give

$$\Delta x \phi_i (v_i)_t + \phi_i (g_{i+1/2} - g_{i-1/2}) = 0, \qquad i > 0 \tag{2.88}$$

$$\frac{\Delta x}{2}\phi_L (v_L)_t + \phi_L (g_{1/2} - \tilde{g}_L) = 0. \tag{2.89}$$

If Equations (2.88) and (2.89) are summed over all $i \geq 0$, then

$$\Delta x \left[\sum_{i>0} \phi_i (v_i)_t + \frac{\phi_L}{2} (v_L)_t \right] - \sum_{i \geq 0} g_{i+1/2} (\phi_{i+1} - \phi_i) - \phi_L \tilde{g}_L = 0. \tag{2.90}$$

Here, the summation by parts rule for spatial derivatives is applied by using the formula $\sum_{i=1}^{N} c_i (d_i - d_{i-1}) = c_N d_N - c_1 d_L - \sum_{i=1}^{N-1} (c_{i+1} - c_i) d_i$ [15]. The use of the summation by parts rule in the discretized form of the governing equation is very similar to the theory of integration by parts for the differential form of the governing equations. Both are intended to remove derivatives from conservative variables and fluxes to a differentiable test function ϕ. Since the semi-discrete numerical scheme is implemented in the governing equation, the summation by parts rule is merely applied to the discretized flux function. Hence

$$\Delta x \left[\sum_{i>0} \phi_i (v_i)_t + \frac{\phi_L}{2} (v_L)_t \right] - \sum_{i \geq 0} \Delta x g_{i+1/2} \frac{\phi_{i+1} - \phi_i}{\Delta x} - \phi_L \tilde{g}_L = 0. \tag{2.91}$$

As the mesh is refined, $\Delta x \to 0$, then equation (2.91) becomes

$$\lim_{\Delta x \to 0} \left[\Delta x \left[\sum_{i>0} \phi_i (v_i)_t + \frac{\phi_L}{2} (v_L)_t \right] - \sum_{i \geq 0} \Delta x g_{i+1/2} \frac{\phi_{i+1} - \phi_i}{\Delta x} - \phi_L \tilde{g}_L \right] = 0. \tag{2.92}$$

Assuming that v_i and $g_{i+1/2}$ are converging to v and g [15], respectively, gives

$$\lim_{\Delta x \to 0} \left[\Delta x \left[\phi_i (v_i)_t + \frac{\phi_L}{2} (v_L)_t \right] \right] \to \int_{x_L}^{x_R} \phi v_t + \phi_L (v_L)_t dx \tag{2.93}$$

and

$$\lim_{\Delta x \to 0} \left[- \sum_{i \geq 0} \Delta x g_{i+1/2} \frac{\phi_{i+1} - \phi_i}{\Delta x} \right] \to - \int_{x_L}^{x_R} g \phi_x dx. \tag{2.94}$$

After substituting equations (2.93) and (2.94) into equation (2.92) and integrating in time, we have

$$\int_{R^+} \int_{x_L}^{x_R} \phi v_t + \phi_L (v_L)_t - g \phi_x dx dt = \int_{R^+} \phi_L \tilde{g}_L dt. \tag{2.95}$$

To finalize the proof, integration by parts in time is applied and the compactness and

smoothness of test function are used to obtain weak solutions

$$\int_{x_L}^{x_R} (\phi_L v_L + \phi v)\, dx - \int_{R^+} \int_{x_L}^{x_R} (v_L(\phi_L)_t + v\phi_t + g\phi_x)\, dxdt = \int_{R^+} \phi_L \tilde{g}_L dt. \quad (2.96)$$

Using the one to one transformation between entropy and conservative variables due to the convexity of the entropy function and substituting the definition of the boundary flux into equation (2.96), the analogous form of weak solution to equation (2.15) is obtained. □

To demonstrate an instructive application of the above theory regarding to derived boundary schemes, the following example is considered.

2.3.2 The Inviscid Burgers' Equation

The Burgers' equation with viscous terms is known as a simplified model of the Navier-Stokes equations, since the both equations have similar convective and viscous terms. It is worth noting that the original Burgers' equation was studied by Burgers with viscous terms [67]. However, the scalar nonlinear Burgers' equation excluding the viscous term is also important, especially for the current work, which is compressible Euler equations of gas dynamics. In this form it is called *inviscid Burgers' equation*,

$$\begin{aligned} v_t + \left(\frac{v^2}{2}\right)_x &= 0, & (2.97)\\ v(x,0) &= v^0(x), & (2.98)\\ v(x_L,t) &= \mathbf{b}_L, & (2.99) \end{aligned}$$

where $g(v) = \left(\frac{v^2}{2}\right)$ is the flux function, $v^0(x)$ and $\mathbf{b}_L$ are the initial and left boundary conditions, respectively. To elucidate the numerical method of the entropy stability discussion for nonlinear systems, the inviscid Burgers' equation is selected with the quadratic entropy function $S = \frac{v^2}{2}$, as in [25]. The advantage of this specific choice of entropy function is not only that it is convex, $S_{vv} = 1 > 0$, but also it is consistent because of having a coherent relationship between the entropy variables and the conservative variables, $\mathbf{s}^T = S_v = v$. With this quadratic entropy function, the corresponding entropy flux and potential pairs are determined by equations (2.36), (2.37) and (2.38)

as

$$Q(v) = \frac{v^3}{3} \tag{2.100}$$

and

$$(\xi, \eta) = \left(\frac{v^2}{2}, \frac{v^3}{6} \right). \tag{2.101}$$

As it was proved in Section 2.2 that to obtain an entropy stability solution, the global entropy change must be bounded in time. Therefore, the inviscid Burgers equation (2.97) is multiplied by the entropy variable from the left

$$v v_t + v \left(\frac{v^2}{2} \right)_x = 0$$

and this equation is integrated in space to give

$$\frac{d}{dt} \int_{x_L}^{x_R} \frac{v^2}{2} dx = - \int_{x_L}^{x_R} v \left(\frac{v^2}{2} \right)_x dx.$$

Applying the method of integration by parts to the RHS of the equation gives

$$\frac{d}{dt} \int_{x_L}^{x_R} \frac{v^2}{2} dx = - \frac{v^3}{3} \Big|_{x_L}^{x_R}. \tag{2.102}$$

Using the definitions of entropy pairs in (2.102), this equation becomes

$$\frac{d}{dt} \int_{x_L}^{x_R} S dx = Q(x_L) - Q(x_R). \tag{2.103}$$

Since the left boundary only is considered, $Q(x_L)$ has to be bounded above to obtain entropy stability, $Q_L = \mathbf{s}_L^T g_L - \eta_L \leq C$, where C is a constant. The proposed numerical algorithm is required to possess entropy conservative and Roe fluxes. The entropy conservative flux needs to satisfy equation (2.47) in Theorem 2.2.1. To this end, the entropy conservative flux is stated as

$$(v_{i+1} - v_i) \hat{g}_{i+1/2} = \frac{v_{i+1}^3 - v_i^3}{6}$$

or

$$\hat{g}_{i+1/2} = \frac{v_{i+1}^2 + v_{i+1} v_i + v_i^2}{6}.$$

Its equivalent form is presented in [25] as

$$\hat{g}_{i+1/2} = \frac{v_{i+1}^2 + v_i^2}{4} - \frac{(v_{i+1} - v_i)^2}{12}. \tag{2.104}$$

The Roe flux is defined as in Section 2.3

$$g_{i+1/2}^{Roe} = \frac{v_{i+1}^2 + v_i^2}{4} - \frac{|\tilde{B}_{i+1/2}|}{2}(v_{i+1} - v_i), \tag{2.105}$$

where the Roe matrix $|\tilde{B}_{i+1/2}|$ satisfies $\Delta g_{i+1/2} = |\tilde{B}_{i+1/2}|\Delta v_{i+1/2}$. Therefore,

$$\begin{aligned} |\tilde{B}_{i+1/2}| &= \left(\frac{v_{i+1}^2 - v_i^2}{2}\right) \Big/ (v_{i+1} - v_i) \\ &= \left|\frac{v_{i+1} + v_i}{2}\right|. \end{aligned}$$

The final form of the Roe flux is

$$g_{i+1/2}^{Roe} = \frac{v_{i+1}^2 + v_i^2}{4} - \frac{|v_{i+1} + v_i|}{4}(v_{i+1} - v_i). \tag{2.106}$$

The left numerical entropy stable boundary flux for Burgers equation is achieved by combining equations (2.104) and (2.106). However, since equation (2.104) satisfies the entropy conservative criteria, there is a need to make a necessary adjustment to obtain a numerical entropy stable boundary flux. This is done by including Roe diffusion matrix and excluding the negative diffusion from entropy conservative flux to obtain a least diffusive entropy stable left boundary flux, i.e., $(v_{i+1} - v_i)^+/12$ (see [25]). Hence

$$\tilde{g}_L = \frac{v_L^2 + \mathbf{b}_L^2}{4} - \left(\frac{|v_L + \mathbf{b}_L|}{4} + \frac{(v_L - \mathbf{b}_L)^+}{12}\right)(v_L - \mathbf{b}_L). \tag{2.107}$$

(See [25, 68]).

2.3.3 The Euler Equations of Gas Dynamics

The main interest of this chapter is to obtain numerical entropy stable solutions for the compressible Euler equations of gas dynamics, which is one of the prototype examples of conservation laws of the form (2.12) in a bounded domain, by satisfying the weakly enforced Dirichlet type boundary conditions. This system of equations is also

known as the inviscid part of the Navier-Stokes equations and neglects the viscosity and heat conduction terms of the Navier-Stokes equations. The compressible Euler equations of gas dynamics are

$$\begin{aligned} \rho_t + (\rho u)_x &= 0 && (2.108)\\ (\rho u)_t + (\rho u^2 + p)_x &= 0 && (2.109)\\ E_t + (u(E+p))_x &= 0. && (2.110) \end{aligned}$$

These can be written in an equivalent vector form as

$$\mathbf{v}_t + \mathbf{g}(\mathbf{v})_x = 0, \tag{2.111}$$

where $\mathbf{v} = [\rho, \rho u, E]^T$ is the solution vector and $\mathbf{g(v)} = [\rho u, \rho u^2 + p, u(E+p)]^T$ is the corresponding flux vector. In general, the total energy (E) comprises the internal and kinetic energies, $E = (\rho u^2)/2 + \rho e$, with $(\rho u^2)/2$ the kinetic energy and ρe the internal energy. Note that since the unknown variables, $\rho,\ u,\ E$ and p are more than the number of equations (2.108), (2.109) and (2.110), there is a need for a fourth auxiliary equation to define the unknown fourth variable, p, from the conservative variables, $\rho,\ u$ and E [15]. This is called the *equation of state* and it is a relationship between the internal energy and pressure [15]. For an ideal gas

$$p = (\gamma - 1)\left(E - \frac{\rho u^2}{2}\right), \tag{2.112}$$

where γ is the ratio of specific heats, $\gamma = 1.4$ for air as air consists primarily of diatomic molecules. Therefore, the internal energy is equal to $\rho e = p/(\gamma - 1)$ from the equation of state.

The numerical algorithm described so far now can be implemented for the Euler equations of gas dynamics to obtain entropy stable numerical solutions using entropy conservative flux and Roe average diffusion matrices. It was proven by Harten [28] that there are several choices of entropy conservative variables, $\mathbf{s}$, due to different entropy functions, $S(v)$. In addition to variety of the entropy functions for the Euler equations, there are also couple of different derived entropy conservative schemes, which

are defined in [38, 39, 68, 69]. The entropy conservative fluxes proposed in [39, 69] are computationally expensive and complicated. On the other hand, the entropy conservative flux proposed in [68] is more practical for numerical simulations and has a lower computational cost, especially compared with Tadmor's entropy conservative flux [39]. The reason that Tadmor's entropy conservative flux is computationally expensive due to the construction of fluxes in integral form in space. Therefore, the practical and cheaper entropy conservative flux proposed in [68] is preferred for the entropy stable scheme in this study. A brief introduction of this entropy conservative flux will be given, which is henceforth called Roe's entropy conservative fluxes [68]. Before giving details of the discrete Roe's entropy conservative flux and Roe diffusion matrices, let us consider the related physical entropy and corresponding entropy functions [68],

$$S = -\frac{\rho F}{\gamma - 1}, \qquad Q = -\frac{\rho u F}{\gamma - 1}, \tag{2.113}$$

where $F = \ln(p\rho^{-\gamma})$ is known as the thermodynamic entropy. The corresponding entropy variables are derived from the entropy function, S, with respect to conservative variables v, such that

$$\mathbf{s}^T = S_v = \left(\frac{\gamma - F}{\gamma - 1} - \frac{\rho u^2}{2p}, \frac{\rho u}{p}, -\frac{\rho}{p}\right). \tag{2.114}$$

This entropy function is the only entropy function that is capable of symmetrizing the full Navier-Stokes equations [30, 39]. Therefore, the study of this chapter can be further extended to study entropy stable schemes for the Navier-Stokes equations.

Roe's entropy conservative flux is introduced by using arithmetic and logarithmic mean functions such that

$$\overline{(m_1, m_2)} = \frac{m_1 + m_2}{2} \tag{2.115}$$

$$(m_1, m_2)^{\ln} = \frac{m_1 - m_2}{\ln(m_1) - \ln(m_2)}. \tag{2.116}$$

Equations (2.115) and (2.116) describe the arithmetic and logarithmic means of two quantities, respectively. Detailed information about the computation algorithm for the logarithmic mean is given in Appendix B. These average quantities were first

proposed by Roe [70] to study the conservation of entropy across contact discontinuity, unlike the other proposed entropy conservative fluxes [39, 69]. This is the another drawback of the entropy conservative fluxes proposed in [39, 69], apart from being computationally expensive and having complicated algorithms. In Roe [70], $\hat{\rho}$ and $\hat{a}$ are averaged by using the logarithmic mean and used in the numerical algorithm to obtain the desired entropy conservation across a contact discontinuity. In addition, the remaining averaged quantities are calculated using the arithmetic mean due to the convenience of its numerical cost. To determine averaged flux quantities, the following components of solution vector $\mathbf{z}$ are used [70]

$$z_1 = \sqrt{\frac{\rho}{p}}, \qquad z_2 = \sqrt{\frac{\rho}{p}}u, \qquad z_3 = \sqrt{\rho p}. \tag{2.117}$$

Then, the average flux quantities are

$$\hat{\rho} = \bar{z_1} z_3^{ln},$$

$$\hat{u} = \frac{\bar{z_2}}{\bar{z_1}},$$

$$\hat{p} = \frac{\bar{z_3}}{\bar{z_1}}.$$

Lastly, the average speed of sound $\hat{a}$ is calculated using [10],

$$\hat{a} = \left(\frac{\gamma \hat{p}_1}{\hat{\rho}}\right)^{1/2}, \tag{2.118}$$

where $\hat{p}_1 = \frac{\gamma+1}{2\gamma}\frac{z_3^{ln}}{z_1^{ln}} + \frac{\gamma-1}{2\gamma}\frac{\bar{z}_3}{\bar{z}_1}$. Then the entropy conservative flux,

$$\hat{g} = \left[\hat{\rho}\hat{u},\ \hat{\rho}\hat{u}^2 + \hat{p},\ \hat{u}\left(\frac{\gamma}{\gamma-1}\hat{p}_1 + \frac{1}{2}\hat{\rho}\hat{u}^2\right)\right]^T \tag{2.119}$$

can be written in an equivalent discretized vector form,

$$\hat{g}_{i+1/2} = \begin{bmatrix} \bar{z_2}_{i+1/2}(z_3^{ln})_{i+1/2} \\ \frac{\bar{z_3}_{i+1/2}}{\bar{z_1}_{i+1/2}} + \frac{\bar{z_2}_{i+1/2}}{\bar{z_1}_{i+1/2}}(\hat{g}_1)_{i+1/2} \\ \frac{1}{2}\frac{\bar{z_2}_{i+1/2}}{\bar{z_1}_{i+1/2}}\left(\frac{\gamma+1}{\gamma-1}\frac{(z_3^{ln})_{i+1/2}}{(z_1^{ln})_{i+1/2}} + (\hat{g}_2)_{i+1/2}\right) \end{bmatrix}. \tag{2.120}$$

Roe's entropy conservative flux is now implemented in the grid interface treatment, the far-field and the wall boundary conditions in Sections 2.3.3.1, 2.3.3.2 and 2.3.3.3 to obtain a bound on the solution, as described in [29].

2.3.3.1 Far-field Boundary Conditions

The far-field boundary condition is intended to allow any in and out going waves to flow through the boundaries smoothly [71]. This boundary condition is called an open or permeable boundary condition. In this section, only left in-going flows are taken into consideration for simplicity. The numerical boundary flux is designed using the entropy conservative flux and Roe matrix. The standard Roe matrix [70] satisfies $g_v = \tilde{R}$ for the Euler equations of gas dynamics in conservative variables. However, the numerical boundary flux is derived using the entropy conservative flux and the Roe average matrix with entropy variables, which is different from the Roe average matrix with conservative variables detailed in Section 2.3. This is because the numerical entropy boundary flux in entropy variables, $\tilde{Q}_L$, has to be bounded to obtain global entropy stability, which yields a bound on the solution for stability. Moreover, the definition of $\tilde{Q}_L$ consists of the numerical boundary flux in entropy variables.

Here, this procedure is reversed to define far-field boundary conditions with respect to conservative variables, instead of entropy variables, due to convenience. To this end, the auxiliary relations in equation (2.65) are adopted for the boundary flux. Initially, the symmetric Roe average Jacobian and symmetric positive definite auxiliary matrices with respect to the physical entropy function are stated as

$$\tilde{G}_L = \int_{-1/2}^{1/2} g_{\mathbf{s}}(\mathbf{s}_{-1/2}(\theta))d\theta \tag{2.121}$$

and

$$\tilde{P}_L = \int_{-1/2}^{1/2} v_{\mathbf{s}}(\mathbf{s}_{-1/2}(\theta))d\theta, \tag{2.122}$$

where $\mathbf{s}_{-1/2}(\theta) = \frac{1}{2}(\mathbf{s}_L + \mathbf{b}_L) + \theta\Delta\mathbf{s}_{-1/2}$ and $\Delta\mathbf{s}_{-1/2} = \mathbf{s}_L - \mathbf{b}_L$ [25, 29]. Then, the matrix $\tilde{R}_L$ is constructed using equations (2.121) and (2.122), so that

$$\tilde{R}_L = \tilde{G}_L\tilde{P}_L^{-1}, \tag{2.123}$$

where $\tilde{P}_L^{-1} = \frac{\mathbf{s}_L - \mathbf{b}(\mathbf{s})_L}{v_L - \mathbf{b}(v)_L}$. Multiplying equation (2.123) by $\tilde{P}_L^{1/2}$ and $\tilde{P}_L^{-1/2}$ from the right and left, respectively, yields

$$\tilde{P}_L^{-1/2} \tilde{R}_L \tilde{P}_L^{1/2} = \tilde{P}_L^{-1/2} \tilde{G}_L \tilde{P}_L^{-1/2}. \tag{2.124}$$

Since the RHS is symmetric due to the physical entropy, so is the left hand side. These can be diagonalized by

$$\tilde{R}_L = \tilde{N}_L \tilde{\Lambda}_L \tilde{N}_L^{-1}, \qquad \tilde{G}_L = \tilde{K}_L \tilde{\tau}_L \tilde{K}_L^T. \tag{2.125}$$

The Roe average Jacobian flux is provided by its absolute value $|\tilde{R}_L| = |\tilde{G}_L| \tilde{P}_L^{-1}$ at the boundary. Therefore, equation (2.125) can be written in the form of

$$|\tilde{R}_L| = \tilde{N}_L |\tilde{\Lambda}_L| \tilde{N}_L^{-1}, \qquad |\tilde{G}_L| = \tilde{K}_L |\tilde{\tau}_L| \tilde{K}_L^T. \tag{2.126}$$

After substituting equations (2.126) in (2.124),

$$\tilde{P}_L^{-1/2} \tilde{N}_L |\tilde{\Lambda}_L| \tilde{N}_L^{-1} \tilde{P}_L^{1/2} = \tilde{P}_L^{-1/2} \tilde{K}_L |\tilde{\tau}_L| \tilde{K}_L^T \tilde{P}_L^{-1/2}$$

and multiplying by $\tilde{P}_L^{1/2}$ from both sides of the above equality, we have

$$\tilde{N}_L |\tilde{\Lambda}_L| \tilde{N}_L^{-1} \tilde{P}_L = \tilde{K}_L |\tilde{\tau}_L| \tilde{K}_L^T.$$

Using the definition $\tilde{P}_L = \frac{v_L - \mathbf{b}(v)_L}{\mathbf{s}_L - \mathbf{b}(\mathbf{s})_L}$ and applying the one-to-one mapping between the entropy and conservative variables, $v \leftrightarrow \mathbf{s}$, we have

$$|\tilde{R}_L|(v_L - \mathbf{b}(v)_L) = |\tilde{G}_L|(\mathbf{s}_L - \mathbf{b}(\mathbf{s})_L). \tag{2.127}$$

The inverse eigenvector matrix equals its transpose such that $\tilde{N}_L^{-1} = \tilde{N}_L^T$ by the symmetric properties of the $\tilde{R}_L$ matrix. $\hat{E}_L$ has the same eigenvectors as $\tilde{R}_L$ and is defined as the diffusion matrix in conservative variables, i.e., $\hat{E}_L = \tilde{N}_L \hat{\Lambda}_L \tilde{N}_L^T$. Consequently, the equivalent form of the numerical boundary flux in entropy variables in equation

(2.69) can be stated in conservative variables as

$$\tilde{g}_L = \frac{g(v_L) + g(\mathbf{b}_L)}{2} - \frac{|\tilde{R}_L| + |\hat{E}_L| + \hat{E}_L}{2}(v_L - \mathbf{b}_L). \tag{2.128}$$

2.3.3.2 Wall Boundary Conditions

In this section, a standard no penetration wall boundary condition is implemented in the numerical scheme for the discretized compressible Euler equations without specifying whether this flow is in or out or sub or supersonic flows. Compared to the far-field boundary conditions, this type of boundary does not allow any flow through the wall. This boundary condition is $\mathbf{v}.\mathbf{n} = 0$ where $\mathbf{n}$ is the outward unit normal to the wall. A mirroring technique is used to negate the normal velocity at the boundary and leaves untouched the remaining flow variables. Hence

$$v_L = (\rho, \rho u, E)^T \tag{2.129}$$

and

$$\mathbf{b}_L = (\rho, -\rho u, E)^T, \tag{2.130}$$

where v_L and $\mathbf{b}_L$ are the solution and the boundary data at the left boundary, respectively. We remark that $v_L = \mathbf{b}_L$ (iff u = 0) if weak enforcement on the boundary is consistent. It is worth noting that the sign of the velocity does not affect the total energy as

$$E = \frac{p}{\gamma - 1} + \frac{\rho u^2}{2}. \tag{2.131}$$

Next, the global entropy bound on the solution is proved by obtaining a bound on the entropy function. Therefore, the entropy flux at the left boundary must also be bounded such that

$$\tilde{Q}_L = \mathbf{s}_L^T \tilde{g}_L - \eta_L \leq C, \tag{2.132}$$

where C is a constant. Here, $\mathbf{s}_L^T$ is an entropy variable with respect to the physical entropy, $S = -\frac{\rho F}{\gamma - 1}$, and is given by

$$\mathbf{s}_L^T = \left(\frac{\gamma - F}{\gamma - 1} - \frac{\rho u^2}{2p}, \frac{\rho u}{p}, -\frac{\rho}{p}\right) \tag{2.133}$$

The corresponding potential function for the same physical entropy is given from the definition (2.38) as in [38],

$$\eta_L = \rho u. \tag{2.134}$$

To obtain the desired bound on the numerical entropy flux, the numerical boundary flux equation is constructed by using the definition in equation (2.128) with the given boundary data (2.130) for the wall boundary condition. The calculation of this wall boundary flux is then commenced with

$$\begin{aligned} \frac{g(v_L)+g(\mathbf{b}_L)}{2} &= \frac{(\rho u, \rho u^2+p, u(E+p))^T + (-\rho u, \rho u^2+p, -u(E+p))^T}{2} \\ &= (0, \rho u^2 + p, 0)^T. \end{aligned} \tag{2.135}$$

Furthermore, the $\hat{E}_L$ and $\tilde{R}_L$ matrices are explicitly defined by using Roe's entropy conservative flux and Roe average diagonalization matrices, respectively [68]. Roe's entropy conservative flux is stated by using the arithmetic average of the flow variable vector, $\bar{\mathbf{z}} = \frac{\mathbf{z}(v_L)+\mathbf{z}(\mathbf{b}_L)}{2}$, between boundary data and solution vectors. Hence,

$$\bar{\mathbf{z}} = \sqrt{\frac{\rho}{p}}(1, 0, p)^T.$$

Then, Roe's entropy conservative flux is derived by substituting this arithmetic average flow variable vector into equation (2.120) so that

$$\hat{g}_L = (0, p, 0)^T. \tag{2.136}$$

Now, $\hat{E}_L$ is calculated from

$$\hat{g}_L - \frac{g(v_L) + g(\mathbf{b}_L)}{2} = -\frac{\hat{E}_L}{2}(v_L - \mathbf{b}_L). \tag{2.137}$$

After substituting equations (2.129), (2.130), (2.135) and (2.136) into equation (2.137), we have [29] :

$$(0, p, 0)^T - (0, \rho u^2 + p, 0)^T = -\frac{\hat{E}_L}{2}(0, 2\rho u, 0)^T$$

and

$$\hat{E}_L = \mathrm{diag}(0, u, 0).$$

In order to calculate the Roe matrix, $|\tilde{R}_L|$, Roe average velocity, the total enthalpy, $h = \frac{E+p}{\rho}$, the density and the speed of sound are needed [63],

$$\tilde{\rho} = \sqrt{\rho_i \rho_{i+1}} \tag{2.138}$$

$$\tilde{u} = \frac{\sqrt{\rho_i}u_i + \sqrt{\rho_{i+1}}u_{i+1}}{\sqrt{\rho_i} + \sqrt{\rho_{i+1}}} \tag{2.139}$$

$$\tilde{h} = \frac{\sqrt{\rho_i}h_i + \sqrt{\rho_{i+1}}h_{i+1}}{\sqrt{\rho_i} + \sqrt{\rho_{i+1}}} \tag{2.140}$$

$$\tilde{a}^2 = (\gamma - 1)\left(\tilde{h} - \frac{1}{2}\tilde{u}^2\right). \tag{2.141}$$

Substituting equations (2.129) and (2.130) into equations (2.138), (2.139), (2.140) and (2.141) gives

$$\tilde{\rho}_L = \rho, \qquad \tilde{u}_L = 0, \qquad \tilde{h}_L = h, \qquad \tilde{a}_L^2 = (\gamma - 1)h. \tag{2.142}$$

Here, the Roe average speed of sound is one of the essential components needed to determine the eigenvalue and eigenvector matrices. In [63], since Roe derived these average values to satisfy $\tilde{R} = g_v$, the Roe average eigenpairs (eigenvalue and corresponding eigenvectors) can be explicitly computed. Therefore, the Roe average eigenvalue matrix is given by

$$\tilde{\Lambda} = \mathrm{diag}(\tilde{u} - \tilde{a}, 0, \tilde{u} + \tilde{a}). \tag{2.143}$$

Since $\tilde{u}_L$ equals zero, the eigenvalue matrix at the left boundary can be written as

$$\tilde{\Lambda}_L = \mathrm{diag}(-a, 0, a). \tag{2.144}$$

The corresponding Roe average eigenvectors are defined by

$$\tilde{N} = \begin{bmatrix} 1 & 1 & 1 \\ \tilde{u} - \tilde{a} & \tilde{u} & \tilde{u} + \tilde{a} \\ \frac{\tilde{a}^2}{\gamma - 1} - \tilde{u}\tilde{a} + \frac{1}{2}\tilde{u}^2 & \frac{1}{2}\tilde{u}^2 & \frac{\tilde{a}^2}{\gamma - 1} + \tilde{u}\tilde{a} + \frac{1}{2}\tilde{u}^2 \end{bmatrix} \tag{2.145}$$

$$\tilde{N}^{-1} = \frac{1}{2\tilde{a}^2}\begin{bmatrix} \frac{\gamma-1}{2}\tilde{u}^2 + \tilde{u}\tilde{a} & -(\gamma-1)\tilde{u} - \tilde{a} & (\gamma - 1) \\ 2\tilde{a}^2 - (\gamma-1)\tilde{u}^2 & 2(\gamma-1)\tilde{u} & -2(\gamma-1) \\ \frac{\gamma-1}{2}\tilde{u}^2 - \tilde{u}\tilde{a} & -(\gamma-1)\tilde{u} + \tilde{a} & (\gamma - 1) \end{bmatrix}, \tag{2.146}$$

where $\tilde{N}$ and $\tilde{N}^{-1}$ are the right and left eigenvector matrices, respectively [68, 72]. After replacing the Roe average values from equation (2.142) into equations (2.145) and (2.146), the following Roe average right and left boundary eigenvectors are obtained

$$\tilde{N}_L = \begin{bmatrix} 1 & 1 & 1 \\ -a & 0 & a \\ h & 0 & h \end{bmatrix} \tag{2.147}$$

$$\tilde{N}_L^{-1} = \begin{bmatrix} 0 & -\frac{1}{2a} & -\frac{1}{2h} \\ 1 & 0 & -\frac{1}{h} \\ 0 & \frac{1}{2a} & \frac{1}{2h} \end{bmatrix}. \tag{2.148}$$

The Roe matrix is then calculated using equations (2.144), (2.147) and (2.148), giving

$$|\tilde{R}_L| = \tilde{N}_L|\tilde{\Lambda}_L|\tilde{N}_L^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & a & \frac{a^2}{h} \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.149}$$

Hence, the left boundary flux, after implementing all flux, matrix and vector values, becomes [29],

$$\begin{aligned} \tilde{g}_L &= \frac{g(v_L) + g(\mathbf{b}_L)}{2} - \frac{|\tilde{R}_L| + |\hat{E}_L| + \hat{E}_L}{2}(v_L - \mathbf{b}_L) \\ &= (0, \rho u^2 + p, 0)^T - (0, 2u^+\rho u, 0)^T - (0, a\rho u, 0)^T \\ &= (0, (u - 2u^+)\rho u - a\rho u + p, 0)^T. \end{aligned}$$

Now, the final form of the boundary entropy flux function, $\tilde{Q}_L$, in equation (2.132) becomes

$$\begin{aligned} \tilde{Q}_L &= \mathbf{s}_L^T g_L - \eta_L \\ &= \frac{\rho u}{p}\left((u - 2u^+)\rho u - a\rho u + p\right) - \rho u \\ &= \frac{\rho^2 u^2}{p}((u - 2u^+) - a). \end{aligned}$$

Since the pressure p and density ρ are assumed to be physically admissible, they must be positive [29, 68, 73]. Hence, the bound on the boundary entropy flux is achieved, $\tilde{Q}_L \leq 0$ for wall boundary conditions. Therefore, the entropy is bounded, even without specifying any kind of inflow or outflow boundary conditions. Note that all interior fluxes are assumed to be entropy stable and satisfy equation (2.48), as stated in Theorem 2.2.1.

2.3.3.3 Grid Interface Condition

The grid interface treatment is practical and less expensive computationally since it allows different convergence rates for each of the subdivided domains via interfaces. In this section, the computational domain is considered as the entire real axis and is split into two sub-domains via an interface at $x = 0$. The domains on the left and right sides of the interface are called the left, $\Omega_L = (-\infty, 0]$, and right, $\Omega_R = [0, \infty)$, computational sub-domains, respectively. The entropy stable finite difference method is then applied for each of these computational sub-domains by using appropriate interface boundary conditions in order to obtain entropy stable solutions. One of the benefits of using the interface treatment is the ability to have different convergence rates for each at the split left and right sub-domains. Therefore, different uniform mesh sizes Δx_L and Δx_R can be employed in both the left and right computational domains, respectively. In addition, the numerical solutions v_L and v_R and corresponding fluxes g_L and g_R of the left and right computational domains can have different numerical values while approaching the common interface point, $x = 0$. It means that different numerical schemes have different interface approximations to the actual solution at the interface, $x = 0$.

The requirements needed to obtain stable numerical solutions using the entropy stable numerical scheme for interface treatment are twofold : *a*) the boundary fluxes at the interface have to satisfy the entropy stability condition and bound the total change of entropy in time and *b*) the conditions of the Lax Wendroff theorem need to hold for the numerical solution to converge to a physically relevant weak solution via the conservative entropy stable method. To this end, the interior and boundary parts of

the governing equation are discretized for the left and right domains as

$$(v_i)_t + \frac{g_{i+1/2} - g_{i-1/2}}{\Delta x_L} = 0, \qquad i = -1, -2, -3 \ldots \tag{2.150}$$

$$(v_L)_t + \frac{\tilde{g}_L - g_{-1/2}}{\Delta x_L/2} = 0, \tag{2.151}$$

where v_i and $g_{i\pm1/2}$ are the numerical solution and flux values of the interior for the left domain, respectively, and v_L and $\tilde{g}_L$ are the boundary solution and flux values at the interface boundary, $x = 0$, of the left domain, respectively. Furthermore, the equations in the right domain are

$$(v_i)_t + \frac{g_{i+1/2} - g_{i-1/2}}{\Delta x_R} = 0, \qquad i = 1, 2, 3 \ldots \tag{2.152}$$

$$(v_R)_t + \frac{g_{1/2} - \tilde{g}_R}{\Delta x_R/2} = 0, \tag{2.153}$$

where v_i and $g_{i\pm1/2}$ are the numerical solution and flux values of the interior for the right domain, respectively. The v_R and $\tilde{g}_R$ represent the boundary solution and flux values at the interface boundary, $x = 0$, of the right domain, respectively. Here, the interior numerical fluxes $g_{i\pm1/2}$ for both domains are assumed to be entropy stable and satisfy Theorem 2.2.1. Then, equations (2.150) and (2.151) for the left domain are multiplied by $\Delta x_L \mathbf{s}_i^T$ and $\Delta x_L \mathbf{s}_L^T/2$, respectively, to give

$$\Delta x_L \mathbf{s}_i^T (v_i)_t + \mathbf{s}_i^T (g_{i+1/2} - g_{i-1/2}) + \frac{\Delta x_L}{2} \mathbf{s}_L^T (v_L)_t + \mathbf{s}_L^T (\tilde{g}_L - g_{-1/2}) = 0.$$

Using the definitions of entropy and entropy flux functions, we have

$$\sum_{i<0} \Delta x_L (S_i)_t + \frac{\Delta x_L}{2} (S_L)_t \leq -\tilde{Q}_L, \tag{2.154}$$

where the numerical entropy flux at the interface boundary is $\tilde{Q}_L = \mathbf{s}_L^T \tilde{g}_L - \eta_L$. Next, the same procedure is followed for the right domain, and the interior and boundary equations (2.152) and (2.153) are multiplied by $\Delta x_R \mathbf{s}_i^T$ and $\Delta x_R \mathbf{s}_R^T/2$, respectively, to yield

$$\Delta x_R \mathbf{s}_i^T (v_i)_t + \mathbf{s}_i^T (g_{i+1/2} - g_{i-1/2}) + \frac{\Delta x_R}{2} \mathbf{s}_R^T (v_R)_t + \mathbf{s}_R^T (g_{1/2} - \tilde{g}_R) = 0.$$

Using the definitions of the entropy pair functions, we similarly have

$$\sum_{i>0} \Delta x_R (S_i)_t + \frac{\Delta x_R}{2}(S_R)_t \leq \tilde{Q}_R, \tag{2.155}$$

where the numerical entropy flux function is $\tilde{Q}_R = \mathbf{s}_R^T \tilde{g}_R - \eta_R$. The total change of the entropy function for the entire computational domain is calculated by summing equations (2.154) and (2.155) from the left and right regions, respectively, giving

$$\sum_{i<0} \Delta x_L (S_i)_t + \sum_{i>0} \Delta x_R (S_i)_t + \frac{\Delta x_L}{2}(S_L)_t + \frac{\Delta x_R}{2}(S_R)_t \leq \mathbf{s}_R^T \tilde{g}_R - \eta_R - \mathbf{s}_L^T \tilde{g}_L + \eta_L. \tag{2.156}$$

The RHS of inequality refers the interface boundary terms, i.e., $BT = \mathbf{s}_R^T \tilde{g}_R - \eta_R - \mathbf{s}_L^T \tilde{g}_L + \eta_L$ and these RHS terms are needed to be bounded to obtain entropy stable solutions. Therefore, interface boundary terms are entropy stable if they satisfy

$$\mathbf{s}_R^T \tilde{g}_R - \eta_R - \mathbf{s}_L^T \tilde{g}_L + \eta_L \leq 0. \tag{2.157}$$

Equality means that the boundary fluxes are entropy conservative. The boundary fluxes are proposed to preserve entropy stability and provide the desired global bound on entropy [29]. Therefore, the entropy stable left boundary flux is

$$\tilde{g}_L = \hat{g}_0 - \frac{|\tilde{B}_0|}{2}(\mathbf{s}_L - \mathbf{b}_L), \tag{2.158}$$

where $\mathbf{b}_L$ is the boundary data and equals the first term from the right region, $\mathbf{b}_L = \mathbf{s}_R$. The opposite right boundary flux is

$$\tilde{g}_R = \hat{g}_0 + \frac{|\tilde{B}_0|}{2}(\mathbf{s}_R - \mathbf{b}_R), \tag{2.159}$$

where $\mathbf{b}_R$ is the boundary data and equals the first term from the left region, $\mathbf{b}_R = \mathbf{s}_L$. Here, $\hat{g}_0$ is a single interface entropy conservative flux, which is calculated from two collocated numerical solutions from each domain and $\tilde{B}_0$ is assumed to be a symmetric positive semi-definite matrix at the interface $x = 0$. Note that the numerical test functions for the interface condition are considered for open boundary (far-field) con-

ditions and solely in-going fluxes are considered to enforce boundary conditions in a weak sense. Therefore, the matrix $\tilde{B}_0$ is calculated by the absolute value of the Roe matrices, $|\tilde{B}_0|$ as in Section 2.3.3.1.

Implementing boundary equations (2.158) and (2.159) into the left hand side of equation (2.157) gives

$$BT = \mathbf{s}_R^T \left(\hat{g}_0 - \frac{|\tilde{B}_0|}{2}(\mathbf{s}_R - \mathbf{s}_L) \right) - \eta_R - \mathbf{s}_L^T \left(\hat{g}_0 + \frac{|\tilde{B}_0|}{2}(\mathbf{s}_L - \mathbf{s}_R) \right) + \eta_L.$$

Here, $(\mathbf{s}_R - \mathbf{s}_L)^T \hat{g}_0 - (\eta_R - \eta_L) = 0$ due to the entropy conservative flux $\hat{g}_0$. Then, the remaining terms satisfy the desired bound on entropy

$$\begin{aligned} BT &= \mathbf{s}_R^T \left(-\frac{|\tilde{B}_0|}{2}(\mathbf{s}_R - \mathbf{s}_L) \right) - \mathbf{s}_L^T \left(\frac{|\tilde{B}_0|}{2}(\mathbf{s}_L - \mathbf{s}_R) \right) \\ &= -(\mathbf{s}_R - \mathbf{s}_L)^T \frac{|\tilde{B}_0|}{2}(\mathbf{s}_R - \mathbf{s}_L) \leq 0. \end{aligned}$$

In [17], the entropy stable Weighted Essentially Non-Oscillatory (WENO) finite difference method was used to simulate compressible flows, i.e. Burgers and Navier-Stokes equations, and the interface fluxes were constructed in a similar manner as in this chapter.

The convergence of the discrete entropy stable solution via a consistent and conservative numerical scheme to a physically relevant weak solution is essential. This can be proved by the well known Lax-Wendroff Theorem for the interface treatment as well. The main objective of this theorem is to provide crucial information about whether the numerical scheme can converge to a true solution in a weak sense. To prove this the left and right parts of the scheme, equations (2.150), (2.151), (2.152) and (2.153), are multiplied by a compactly supported and bounded test function $\phi(x)$, which is projected onto a grid as $\phi(x_i) = \phi_i$, $i = \ldots, -1, 0, 1, \ldots$,

$$\phi_i (v_i)_t + \phi_i \left(\frac{g_{i+1/2} - g_{i-1/2}}{\Delta x_R} \right) + \frac{\phi_0}{2}(v_R)_t + \frac{\phi_0}{2} \left(\frac{g_{1/2} - \tilde{g}_R}{\Delta x_R/2} \right) = 0$$

and

$$\phi_i (v_i)_t + \phi_i \left(\frac{g_{i+1/2} - g_{i-1/2}}{\Delta x_L} \right) + \frac{\phi_0}{2}(v_L)_t + \frac{\phi_0}{2} \left(\frac{\tilde{g}_L - g_{-1/2}}{\Delta x_L/2} \right) = 0.$$

After summing these equations and integrating in time, we have

$$\int_{R^+} \left(\sum_{i\neq 0} \phi_i (v_i)_t + \frac{\phi_0}{2}((v_L)_t + (v_R)_t) - \sum_{i<0} \frac{\phi_{i+1} - \phi_i}{\Delta x_L} g_{i+1/2} + \phi_0 \tilde{g}_L \right.$$
$$\left. - \sum_{i>0} \frac{\phi_{i+1} - \phi_i}{\Delta x_R} g_{i+1/2} - \phi_0 \tilde{g}_R \right) dt = 0.$$

It is assumed that the numerical solution at the interface v_0 is the mean of the solutions from the right and left domains, i.e., $v_0 = \frac{v_L + v_L}{2}$. Then, using the definitions of the left and right numerical boundary fluxes $\tilde{g}_{L,R}$ from equations (2.158), and (2.159) and integrating by parts in time, we have

$$\sum_{i\in R} \phi_i(0) v_i(0) - \int_{R^+} \left(\sum_{i\in R} (\phi_i)_t v_i + \sum_{i<0} \frac{\phi_{i+1} - \phi_i}{\Delta x_L} g_{i+1/2} + \sum_{i>0} \frac{\phi_{i+1} - \phi_i}{\Delta x_R} g_{i+1/2} \right) dt = 0.$$

The same procedure is used to show that the conditions of the Lax Wendroff Theorem are satisfied as in Section 2.3.1, and this leads to convergence to the desired weak solution of the governing equation for the interface treatment

$$\int_R \phi(x,0) v(x,0) dx - \int_{R^+} \int_R (\phi_t v + \phi_x g) = 0. \tag{2.160}$$

2.3.3.4 Diffusion Limiter

It has already been stated that entropy is conserved in smooth regions and dissipated at discontinuities known as shock waves. The entropy stability scheme was developed to mimic this physical requirement. Therefore, the entropy conservative and stable fluxes were derived for smooth and non-smooth regions, respectively. In this section, the limiter function is implemented into the numerical entropy stable fluxes to obtain more accurate results by maintaining the size of the extra numerical Roe diffusion matrix as low as possible. A too large artificial diffusion might lead to superfluous dissipation around shocks and less accurate solutions in the smooth regions. In general, the least diffusive numerical schemes converge towards more accurate solutions for flows including more complicated physical features, e.g., turbulence, vortices. These features persist for a long time and not be dissipated by numerical damping. Here, the least

diffusive entropy conservative flux and numerical artificial Roe diffusion matrix are derived together as the numerical flux function [29]

$$g_{i+1/2} = \hat{g}_{i+1/2} - \frac{\alpha_{i+1/2}|\tilde{R}_{i+1/2}|}{2}(v_{i+1} - v_i), \tag{2.161}$$

where $\hat{g}_{i+1/2}$ is the entropy conservative flux and $\tilde{R}_{i+1/2}$ is the artificial Roe diffusion matrix. Also, $\alpha_{i+1/2}$ is the limiter scalar function, $\alpha_{i+1/2} \in [0, 1]$. It was also shown in Section 2.3 that the entropy conservative flux can be written in the equivalent form

$$\hat{g}_{i+1/2} = \frac{g_i + g_{i+1}}{2} - \frac{\hat{E}_{i+1/2}}{2}(v_{i+1} - v_i). \tag{2.162}$$

After substituting equation (2.162) into equation (2.161), we have

$$g_{i+1/2} = \frac{g_i + g_{i+1}}{2} - \frac{\hat{E}_{i+1/2} + \alpha_{i+1/2}|\tilde{R}_{i+1/2}|}{2}(v_{i+1} - v_i). \tag{2.163}$$

The detailed description of the limiter scalar function results from the following entropy stability criteria. First, to obtain entropy stable solutions Theorem 2.2.1 has to be satisfied by the numerical flux functions. If $\alpha_{i+1/2}$ equals 1, the flux function is entropy stable and the stability of the solution is achieved. This was also proved in Section 2.3. On the other hand, if $\alpha_{i+1/2}$ equals zero, then the flux function is entropy conservative and stability results. Since fluxes satisfy the entropy stability condition, it is difficult to determine where to add the extra numerical diffusion matrix into the system. For this reason the boundary flux is solely evaluated with a non-diffusive central scheme by omitting the $\hat{E}_{i+1/2}$ and $\tilde{R}_{i+1/2}$ matrices. In [34], it was found that non-diffusive central difference schemes fail to preserve continuity when shocks exist in the system and the scheme becomes unable to capture and mimic this physical feature. To this end, only the non-diffusive central scheme of the flux is substituted into the entropy stability condition to localize the regions for adding the artificial diffusion matrix. Hence

$$\text{if} \quad (\mathbf{s}_{i+1} - \mathbf{s}_i)^T \left(\frac{g_{i+1} + g_i}{2} \right) - (\eta_{i+1} - \eta_i) \leq 0, \qquad \text{region is smooth} \tag{2.164}$$

$$\text{if} \quad (\mathbf{s}_{i+1} - \mathbf{s}_i)^T \left(\frac{g_{i+1} + g_i}{2} \right) - (\eta_{i+1} - \eta_i) > 0, \qquad \text{region is non-smooth.} \tag{2.165}$$

It is clear that for non-smooth regions, the limiter scalar function $\alpha_{i+1/2}$ equals 1. On the other hand, the scalar function is assumed to be zero for smooth regions, which is true in general. However, the amount of the numerical artificial diffusion matrix is modified around the neighborhood of a discontinuity to smooth the behavior in the numerical solution by simple averaging [29]. To do this, the binary value $\mu \in \{0, 1\}$ is introduced such that [29]

$$\mu_{i+1/2} = \begin{cases} 0, & \text{for smooth regions} \\ 1, & \text{for non-smooth regions.} \end{cases}$$

and $\alpha_{i+1/2}$ becomes

$$\alpha_{i+1/2} = \begin{cases} \frac{\mu_{i-1/2}+\mu_{i+3/2}}{2}, & \text{for smooth regions} \\ \mu_{i+1/2} = 1, & \text{for non-smooth regions.} \end{cases}$$

It is worth noting that this limiter process is applied to the numerical fluxes for no-penetration wall and far-field boundary conditions and interface treatment and it preserves stability. The robustness of these derived entropy stable boundary schemes is demonstrated with numerical simulations in Section 2.4, as described in [29].

2.4 Numerical Computations

In this Chapter, the Euler equation of compressible gas dynamics (2.39), (2.52) and (2.163) are semi-discretized using the fourth order Runge-Kutta scheme and the entropy stable finite difference scheme in time and in space, respectively based on the physical entropy pair

$$S = -\frac{\rho F}{\gamma - 1}, \qquad Q = -\frac{\rho u F}{\gamma - 1}.$$

To illustrate the robustness and accuracy of the entropy stable boundary numerical scheme, three different numerical simulations are presented to test the interface treatment, far-field and wall boundary conditions. Throughout the numerical simulations, uniform mesh sizes Δx are used and the time step Δt is adjusted to satisfy the CFL condition. The detailed explanation of the CFL condition can be found in Chapter 4.

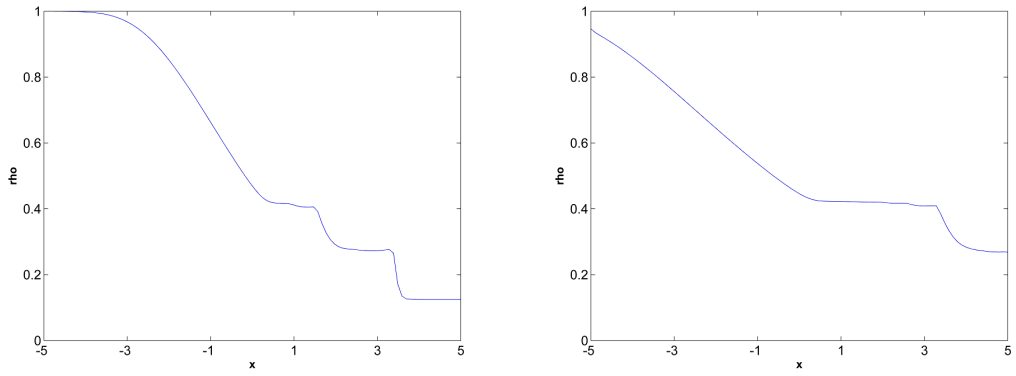
2.4.1 Sod's Shock Tube Test

This test case was proposed in [54] and it was based on several simulations of different numerical methods and comparison of the results from these simulations with the exact solution of the Riemann problem [10]. The compressible time dependent Euler equations of gas dynamics are studied as a canonical example of conservation laws. Therefore, the piecewise constant initial conditions proposed by Sod can be applied to the governing equations and the accuracy of the numerical solution can be tested.

The Euler equation of gas dynamics is solved with the following piecewise constant initial data on the domain $\Omega = [-5, 5]$

$$\begin{aligned} (\rho, u, p) &= (1, 0, 1) \quad x < 0 \\ (\rho, u, p) &= (0.125, 0, 0.1) \quad x \geq 0. \end{aligned}$$

This initial discontinuity develops into a left moving rarefaction wave, right moving shock wave and right going contact discontinuity [10]. All of these flow features and the interaction of the shock wave with the boundary are illustrated by the density using the far-field boundary condition. In general this test case is run until $Time = 2.0$ and is shown in Figures (2.1a) and (2.2a) for 100 and 200 grid points [29]. Similar

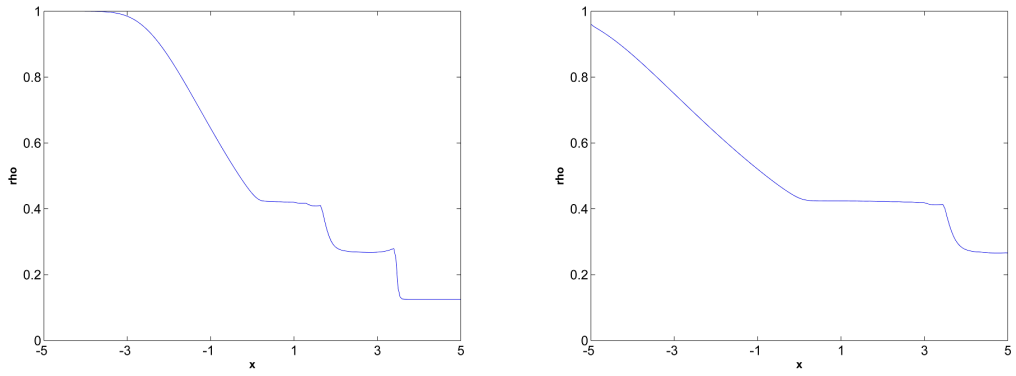


(a) **(b)**

Figure 2.1: Numerical solution of ρ with 100 grid points **(a)** at $Time = 2.0$, **(b)** at $Time = 4.0$.

numerical simulations can be found in [74]. However, since the intention is to test

the behavior of the shock wave under the far-field boundary condition, the simulation is run to $Time = 4.0$. It is clear that the shock wave passes through the boundary without any stability issues or reflection, as seen in Figures (2.1b) and (2.2b) [29]. Analogous simulations were run for the entropy stable boundary scheme with the far-



(a) **(b)**

Figure 2.2: Numerical solution of ρ with 200 grid points **(a)** at $Time = 2.0$, **(b)** at $Time = 4.0$.

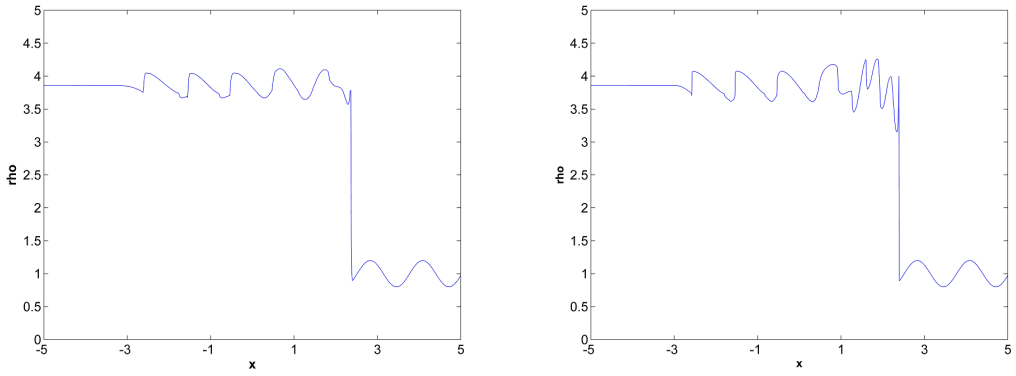
field boundary condition in [35]. However, the proposed study in [35] was based on homogeneous entropy pairs, which is different to the one proposed in this Chapter.

2.4.2 Test for Shock-Entropy Wave Interaction

This test case is designed for studying the interaction between strong shock waves of Mach number of 3 and an entropy wave and was proposed in [74]. The numerical simulations were run subject to the following initial condition,

$$\begin{aligned}(\rho, u, p) &= (3.857143, 2.629369, 10.33333) \quad x < -4 \\ (\rho, u, p) &= (1 + \epsilon \sin(5x), 0, 1) \quad x \geq -4,\end{aligned}$$

where $\epsilon = 0.2$. The computational domain is the same as in the previous test case, i.e., $\Omega = [-5, 5]$. The numerical results are illustrated by the density. The experiments conducted in [74] were limited to $Time = 1.8$. The same case is demonstrated with 800 and 3000 grid points in Figures 2.3a and 2.3b [29]. The results obtained here can then be compared with existing similar studies in the literature [35, 41, 74]. As we mainly interested in how the numerical scheme copes with the interaction of the shock wave



(a) **(b)**

Figure 2.3: Numerical solution of ρ at $Time = 1.8$ **(a)** with 800 grid points, **(b)** with 3000 grid points.

with the boundary, the resolution of the numerical solutions is not our main concern. Therefore, the numerical test was run with 400 grid points at three different times $Time = 1.8, 2.5$ and 2.8 (see Figure 2.4). In Figures 2.4a and 2.4b, the density was calculated before the shock wave arrives at the boundary and in Figure 2.4c the shock wave has passed through the boundary at $Time = 2.8$ without any difficulty. The shock then passes through the boundary without any stability problem, which is the desired result.

2.4.3 Test for Interaction of Shock and Wall

This test case, which is also known as the Woodward-Colella test case, is standard for the compressible Euler equations of gas dynamics [75]. This test case was originally constructed for testing the ability of a scheme to cope with strong shock waves. However, instead of the collision of strong shock waves, the main interest is to observe the reflection of strong shock waves from a wall and their interaction with each other. This test is a very severe test of the robustness of the proposed boundary scheme for no-penetration of a wall, as detailed in Section 2.3.3.2. To apply this test to the proposed entropy stable boundary scheme, the initial data given in [74, 75] and

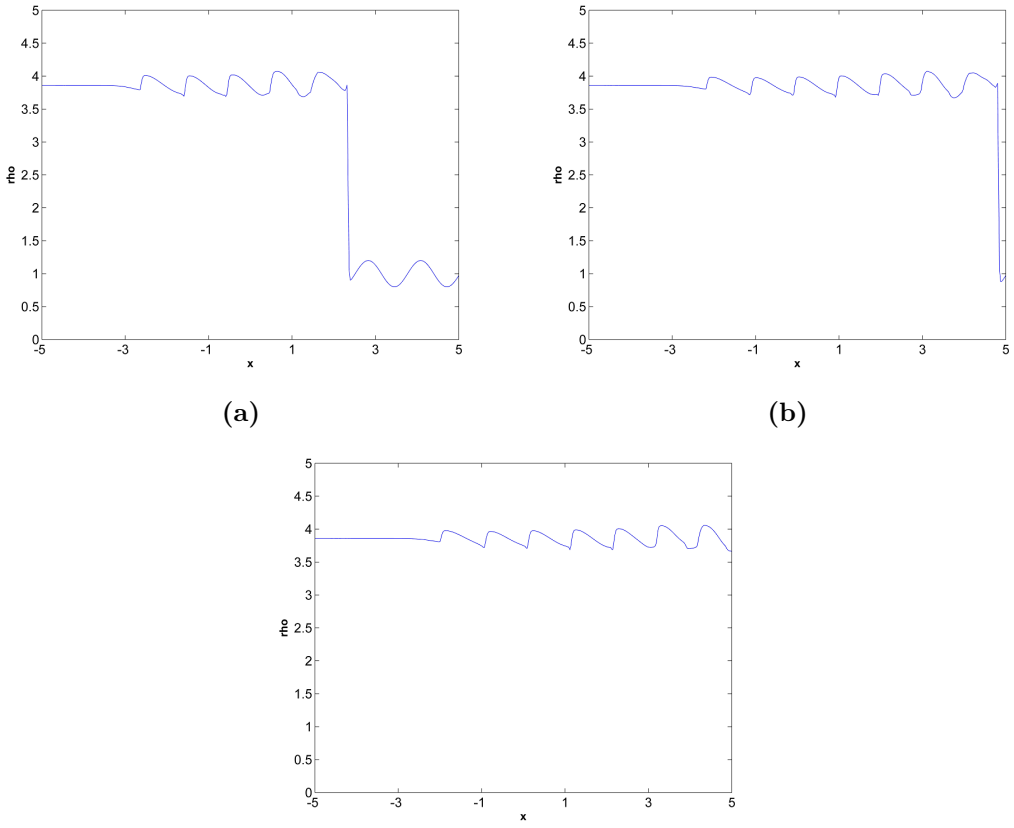
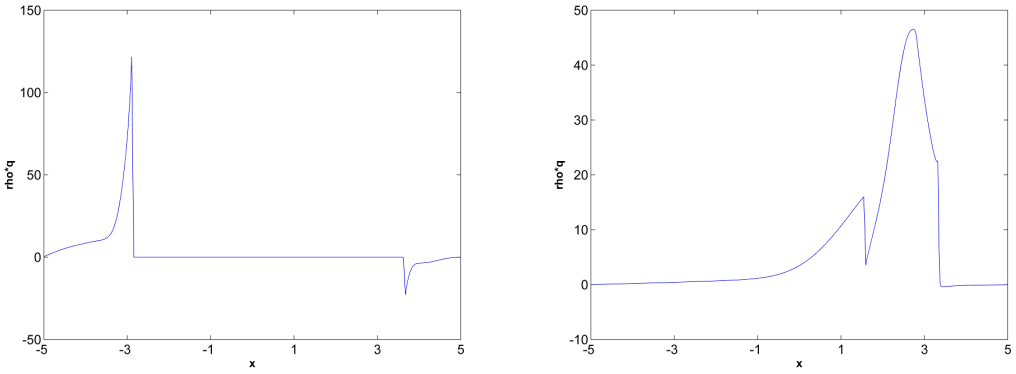


Figure 2.4: Numerical solution of ρ with 400 grid points **(a)** at $Time = 1.8$, **(b)** at $Time = 2.5$, **(c)** at $Time = 2.8$.

the computational domain $\Omega = [-5, 5]$ are used:

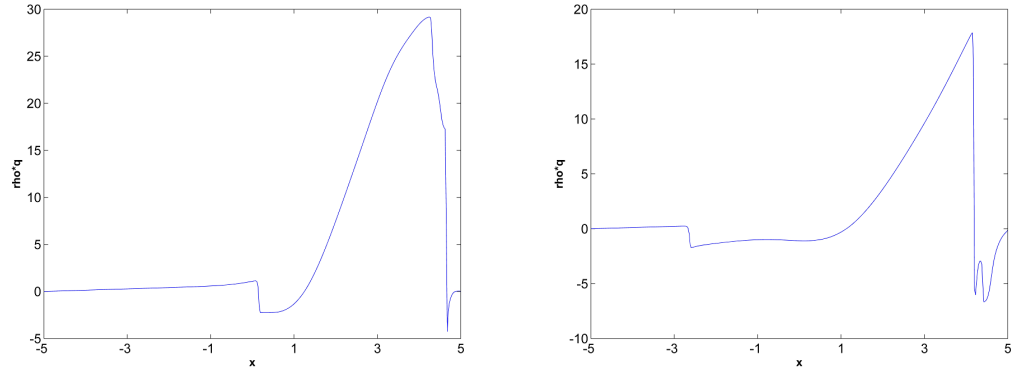
$$\begin{aligned}(\rho, u, p) &= (1, 0, 1000) \quad 0 \leq x < 0.1 \\ (\rho, u, p) &= (1, 0, 0.01) \quad 0.1 \leq x < 0.9 \\ (\rho, u, p) &= (1, 0, 100) \quad 0.9 \leq x < 1.\end{aligned}$$

The boundary data $\mathbf{b}_{Lx=-5}(t)$ and $\mathbf{b}_{Rx=5}(t)$ are defined by mirroring the velocities of the solution $v_{\{x=-5,5\}}(t)$ (see Section 2.3.3.2). The numerical simulations were first run with 400 grid points until $Time = 0.05, 0.38$. This is before the shock wave reaches the right boundary, as seen in Figures 2.5a and 2.5b. Then, the simulations were run until $Time = 0.5, 0.6$ so that the shock interacted with the wall. In Figures 2.6a and 2.6b, it is observed that the boundary scheme is successful in allowing the right moving shock wave to reach and reflect from the wall. Consequently, it can be seen that there is no



(a) **(b)**

Figure 2.5: Numerical solution of ρu with 400 grid points **(a)** at $Time = 0.05$, **(b)** at $Time = 0.38$.



(a) **(b)**

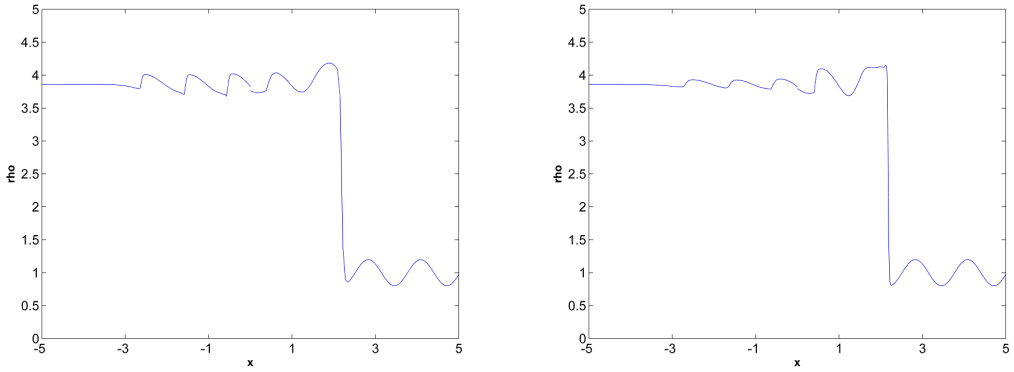
Figure 2.6: Numerical solution of ρu with 400 grid points **(a)** at $Time = 0.5$, **(b)** at $Time = 0.6$.

stability problem for the solution due to the boundary. A similar study with different entropy (homogeneous entropy) can be found in [35].

2.4.4 Test for Grid Interface Treatment of Shock

In this section, the shock entropy wave experiment, which was explained in Section 2.4.2, is computed for testing the robustness of the treatment of the interface [29]. The numerical simulations are depicted using the density at $Time = 1.8$ with different resolutions for each of the left and right hand domains. In Figure 2.7a, the density is depicted using 200 and 80 grid points for the left and right domains, respectively. In addition, the resolution of the numerical solution was changed for each domain by computing the solution in the left and right domains with 80 and 200 grid points, as

shown in Figure 2.7b. It can be seen from the results presented in Figure 2.7 that the solution is more diffusive for the lower resolution domain, as expected, and there is no problem with the stability for the shock passing the grid interface at $x = 0$ [29]. Next, a comparison is made between the interface and the corresponding non-interface



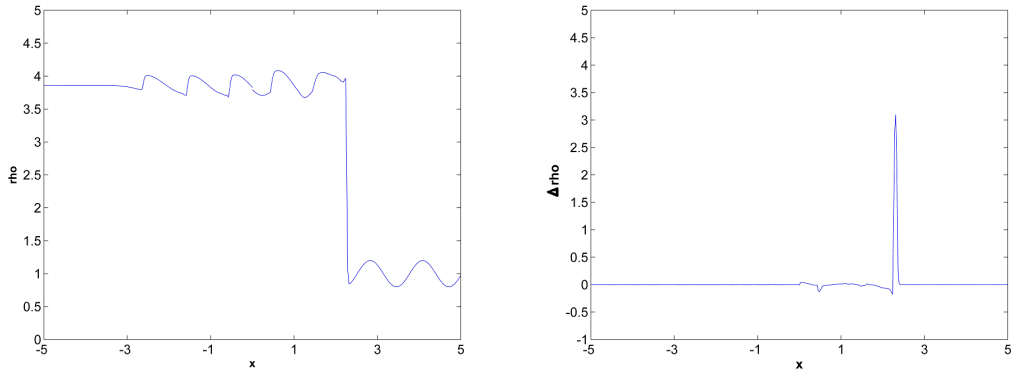
(a) **(b)**

Figure 2.7: Numerical solution of ρ at $Time = 1.8$ **(a)** with 200 and 80 grid points for the left and right domains, **(b)** with 80 and 200 grid points for the left and right domains.

treatment based on a pointwise difference between them. In Figure 2.8a, a numerical solution with 200 grid points in both the left and right hand domains is shown at $Time = 1.8$ [29]. In Figure 2.8b, a comparison between the numerical solutions for the interface scheme with 200 grid points and the corresponding non-interface scheme with 399 points is depicted via their pointwise difference [29]. There is a peak which is a one cell difference at the shock location. Since entropy stability is satisfied and the weak solution is obtained by a conservative numerical scheme, then the interface treatment still preserves stability even if with the existence of a one cell difference. However, this peak indicates that a weak solution cannot be in L^∞ as Δx tends to zero. Nonetheless, convergence in L^2 can still be obtained.

2.4.5 A Fourth Order Entropy Stable Scheme

In this section, the intention is to show that the entropy stable boundary numerical scheme can also be applied with higher order numerical schemes. In [41], a fourth order entropy stable scheme with dual-diffusion was derived for the Euler equation of gas dynamics with homogeneous entropy. It should be emphasized that fourth order is



(a) **(b)**

Figure 2.8: Numerical solution of ρ at $Time = 1.8$ **(a)** with 200 grid points for both domains, **(b)** pointwise difference.

the order of the smooth solutions. The scheme proposed in [41] was modified according to the physical entropy, as explained in Section 2.3.3. This can be done with a slight modification of the boundary flux and the limiter function in order to satisfy the entropy stability criteria (see Theorem 2.2.1) for the physical entropy. However, since the intention is to test the boundary condition, the shock limiter will be left in its original form, as in [41]. It is worth noting that even if the system contains more than enough diffusion, it still leads to an entropy stable solution under the conditions of the entropy stability condition in Theorem 2.2.1. The only drawback of adding the extra diffusion to the system, especially for smooth regions, is that the numerical results would be highly diffusive and so lead to a poor approximation to the solution of the governing PDEs. For this reason, the limiter plays an essential role in dampening the oscillations in the vicinity of shock waves [61].

Before detailing the results from the numerical simulations, a modified version of the fourth order scheme is presented as follows

$$(v_i)_t + \frac{g^{4th}_{i+1/2} - g^{4th}_{i-1/2}}{w_i \Delta x} = 0 \tag{2.166}$$

$$(v_L)_t + \frac{g^{4th}_{1/2} - \tilde{g}^{4th}_{L}}{w_L \Delta x} = \mathbb{S}_L, \tag{2.167}$$

where

$$
\begin{aligned}
g^{4th}_{i+1/2} &= -\frac{1}{12}(g_{i+2} + g_i + D_{i+1}(\mathbf{s}_{i+2} - \mathbf{s}_i)) \\
&\quad + \frac{8}{12}(g_{i+1} + g_i - D_{i+1/2}(\mathbf{s}_{i+1} - \mathbf{s}_i)) \\
&\quad - \frac{1}{12}(g_{i+1} - g_{i-1} + D_i(s_{i+1} - \mathbf{s}_{i-1}))
\end{aligned} \tag{2.168}
$$

and $w_i > 0$ are weights depending on the discrete norm [29]. Here, $\mathbb{S}_L$ is called the Simultaneous Approximation Term, as in [41], and is defined by $\mathbb{S}_L = -\frac{1}{w_L \Delta x} A_L^+(\mathbf{s}_L - \mathbf{b}_L)$, where A_L is the Jacobian of the flux g at $x = x_L$, i.e., $\frac{\partial g}{\partial \mathbf{s}_L} = A_L$, and the plus sign refers to the part with positive eigenvalues. $\mathbb{S}_L$ is used for enforcing the boundary conditions corresponding to in going flows in the weak sense. The boundary flux is modified by making changes to $\mathbb{S}_L$ on the RHS of the discretized boundary scheme in equation (2.167). It then becomes

$$
\mathbb{S}_L = -\frac{1}{w_L \Delta x}(|\tilde{G}_L| + \hat{D}_L + |\hat{D}_L|)(\mathbf{s}_L - \mathbf{b}_L), \tag{2.169}
$$

where $\tilde{G}_L$ is the Roe average Jacobian matrix between the boundary data and the solution variables, i.e., $\tilde{G}_L(\mathbf{s}_L - \mathbf{b}_L) = g(\mathbf{s}_L) - g(\mathbf{b}_L)$. Then, the modified form of the boundary scheme for the fourth order scheme is [29]

$$
(v_L)_t + \frac{g^{4th}_{1/2} - \tilde{g}^{4th}_L}{w_L \Delta x} = 0, \tag{2.170}
$$

where

$$
\tilde{g}^{4th}_L = \frac{g(\mathbf{s}_L) + g(\mathbf{b}_L)}{2} - \frac{|\tilde{G}_L| + \hat{D}_L + |\hat{D}_L|}{2}(\mathbf{s}_L - \mathbf{b}_L). \tag{2.171}
$$

To obtain a global entropy estimate with this modified boundary scheme, the interior and boundary schemes in equations (2.166) and (2.167) are multiplied by $w_i \Delta x \mathbf{s}_i^T$ and $w_L \Delta x \mathbf{s}_L^T$, respectively. Following the argument in [41], the diffusion matrices $D_i > D_i^{ec}$ and $D_{i+1/2} > D_{i+1/2}^{ec}$ are assumed to satisfy the entropy stability condition and to have more diffusion than the entropy conservative fluxes. Then, the local numerical entropy

inequalities for the interior and boundary schemes are

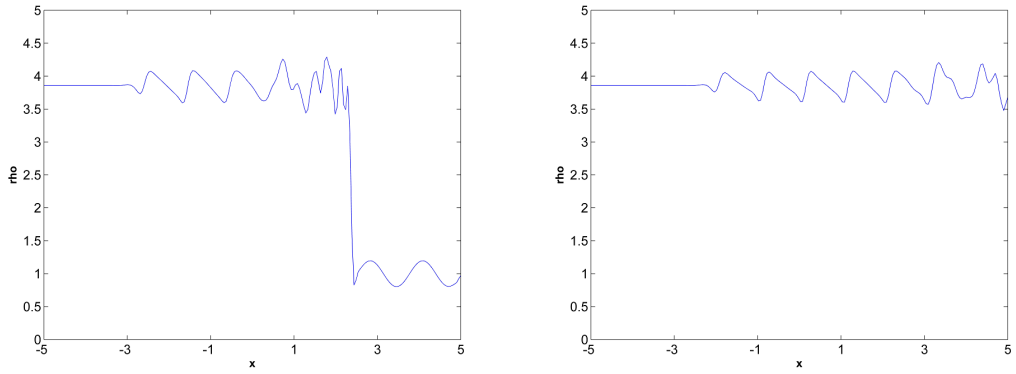
$$\begin{aligned} w_i \Delta x (S_i)_t + Q^{4th}_{i+1/2} - Q^{4th}_{i-1/2} &\leq 0 \\ w_L \Delta x (S_L)_t + Q^{4th}_{1/2} - \tilde{Q}^{4th}_{L} &\leq 0. \end{aligned}$$

After adding the above equations, the following inequality is obtained

$$\sum_{i \geq 0} w_i \Delta x (S_i)_t - \tilde{Q}^{4th}_{L} \leq 0.$$

It is clear from the theory in this Chapter that $\tilde{Q}^{4th}_{L}$ has to be bounded to satisfy the global entropy estimate. Since the numerical entropy flux at the left boundary $\tilde{Q}^{4th}_{L}$ is defined by the same entropy stable boundary flux as in Section 2.3, i.e., $\tilde{Q}_L = \mathbf{s}_L \tilde{g}^{4th}_{L} - \eta_L = \tilde{Q}^{4th}_{L}$, the proof of a global bound on the entropy is as explained in Section 2.3.

Next, we show numerical computations to illustrate the efficacy of this modified entropy stable boundary scheme. We use the shock entropy wave experiment case of Section 2.4.2. In Figures 2.9a and 2.9b numerical results are presented to illustrate the robustness and accuracy of the scheme of this work [29]. It can be visualized from Figure 2.9 that the shock passes through the boundary with no reflections. There are some advantages of using higher order schemes to obtain more accurate approximations to the solution as compared with lower order schemes. For instance, less computational cost and lower memory usage (with less grid points) can result with the same accuracy when the order of the scheme is increased. In comparison with Figures 2.3b and 2.9a, the first result requires more computational cost and uses more memory to obtain the analogous resolutions.



(a) **(b)**

Figure 2.9: Numerical solution of ρu with 200 grid points **(a)** at $Time = 1.8$, **(b)** at $Time = 2.8$.

Chapter 3

Method of Manufactured Solutions(MMS)

The objective of this dissertation is to find numerical solutions of non-linear hyperbolic systems of PDEs, which mostly appear in computational fluid dynamics (CFD) field. The fundamental aim of CFD is to employ numerical methods and algorithms to solve and analyze fluid dynamic problems. To this end, one has to be sure that the computer code designed to solve a set of partial differential equations is accurate. Therefore, one of the essential points to check the accuracy of an existing code is ensuring that verification and validation procedures have been provided [52, 53]. While verification and validation can be used interchangeably in everyday life, this is not the case in programming language. The latter was expressed informally by B.Boehm [51] via the following questions:

"**Verification :** Are we building the product right ?

Validation : Are we building the right product ?"

It is worth noting that this chapter only deals with the verification step while the validation part will be discussed in Chapter 4. There are several different ways to confirm the verification of written computer codes. Method of Manufactured Solutions (MMS) [52, 53] was introduced as one of the rigorous verification tools to find coding bugs and

to estimate order of accuracy $O(\Delta x^q)$, where $q > 0$ is the formal order of accuracy of developed numerical method. This invaluable tool can easily be implemented into a computer code and is an efficient option to prove that code is bug free. The main reason of the use of MMS is to test the accuracy of computer code by using manufactured solutions. Such manufactured solutions are perfectly convenient solutions for nonlinear PDEs, since nearly all non-linear PDEs have no exact solutions [76]. This method is based on evolving a *priori* known analytical solution of the governing equations. The MMS procedure has been used successfully in different application areas such as fluid dynamics [77, 78], heat transfer [79, 80], and fluid-structure interaction [81]. In the process of MMS, the essence of the idea is to insert manufactured solutions into the governing equations. This is done by computing convergence rate (order of accuracy) for spatial and temporal discretization of numerical schemes. To ensure that the results from numerical simulations are accurate, formal and observed (calculated) convergence rates must match. When implementing MMS procedure into a set of computer code accurately, the following steps are needed to be followed [82]:

Step (1) The type of the governing system of PDEs is determined.

Step (2) The discretization method with formal order of accuracy is selected to solve continuum PDEs numerically.

Step (3) The form of a physically convenient manufactured solution is constructed for the governing equations.

Step (4) The manufactured solution is substituted into the governing system of PDEs to generate new or modified analytic source terms.

Step (5) The manufactured solutions with new source terms are implemented into the governing equations. Later, they are discretized and solved for multiple grid refinements.

Step (6) The global discretization error in the numerical solutions is evaluated for every different mesh sizes.

Step (7) The observed order of accuracy is calculated and then compared with the formal order of accuracy.

If the observed and formal order of accuracy values are matched as explained in Step 7, the numerical code does not need debugging and can be used for solving the governing system of PDEs. It should be noted that although the MMS procedure is used only for proving order of accuracy, this method does not test the robustness, efficiency and stability related issues for a given numerical method [78]. The system of PDEs is semi-discretized in our case; therefore, the focus is given to compute the spatial convergence rate using the MMS procedure. However, there are some restrictions for the selection of the manufactured solutions which are implemented into the governing system of PDEs as follows [82]:

* The chosen manufactured analytical solution must be continuous and differentiable at least up to the order of the corresponding terms in the governing equations.
* To avoid any numerical complications, manufactured solutions must not violate any physically admissible assumptions, for example, the estimated pressure, density and temperature values for gases in the designed system must be positive [68, 73].
* Since the major focus of this procedure is to test the convergence rate of the governing equations, Taylor series expansions of the manufactured solutions must include terms up to the expected order of numerical scheme.

In the following sections, we applied the MMS method to the designed code for its verification.

3.1 Implementation of MMS into Compressible Euler Equations of Gas Dynamics in 2D

3.1.1 Numerical Formulation

The application of the MMS procedure is tested on the two-dimensional systems of non-linear hyperbolic conservation laws

$$u_t + f(u)_x + g(u)_y = 0. \tag{3.1}$$

One of the canonical examples of (3.1) is the Euler equations of gas dynamics being an inviscid compressible flow equation. The following matrix form of the Euler equations

represents the conservation of mass, momentum and energy

$$\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}_t + \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{bmatrix}_x + \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{bmatrix}_y = 0, \tag{3.2}$$

where ρ is density, p is pressure, u and v are the velocities in x and y directions, respectively, and E is the total energy. The stated solution vector $u = (\rho, \rho u, \rho v, E)^T$ is then calculated from the corresponding flux vectors $f(u) = (\rho u, \rho u^2 + p, \rho uv, u(E+p))^T$ and $g(u) = (\rho v, \rho uv, \rho v^2 + p, v(E+p))^T$. The pressure p is determined from the equation of state for an ideal gas in 2D [15],

$$p = (\gamma - 1)\left(E - \frac{\rho(u^2 + v^2)}{2}\right) \tag{3.3}$$

where $\gamma = c_p/c_v$ is the ratio of specific heats, and c_p and c_v are specific heat values at constant pressure and volume, respectively. To suppress numerical oscillations caused by numerical discontinuities, we will modify Euler equation (3.1), by adding adequate amount of artificial viscosity terms [61]. Then, the modified Euler equations becomes

$$u_t + f(u)_x + g(u)_y = \epsilon_1 u_{xx} + \epsilon_2 u_{yy} \tag{3.4}$$

which can be written in a conservative form

$$u_t + (f(u) - \epsilon_1 u_x)_x + (g(u) - \epsilon_2 u_y)_y = 0,$$

where ϵ_1 and ϵ_2 are the constants ranging from 0 and 1, $0 \leq \epsilon_{1,2} \ll 1$.

3.1.2 Analytical Manufactured Solution of the System of PDEs

The following manufactured analytical solutions based on sine and cosine functions are selected for testing the order of accuracy of the interior spatial discretization of the

compressible Euler equations;

$$\begin{aligned} \rho &= \sin(2\pi x)\sin(3\pi y) \\ u &= \sin(\pi x)\cos(\pi y) \\ v &= -\cos(\pi x)\sin(\pi y) \\ p &= \sin^2(2\pi x)\sin^3(3\pi y). \end{aligned} \tag{3.5}$$

These solutions are substituted into the two-dimensional Euler equations (3.4) to determine new or modified source terms for each conserved quantity of mass, momentum, and energy. Therefore, the Euler equations including artificial viscosity terms can then be modified using new additional source terms so that Equation (3.5) is a solution of the modified equations.

3.1.3 Source Terms of Conservation of Mass, Momentum, and Energy Equations

Let's start with generating source terms for the conservation of mass equation by substituting the manufactured solutions (3.5) into equation (3.4). The additional terms required for equation (3.5) to be a solution, do not have to be physical [76].

$$\rho_t + (\rho u)_x + (\rho v)_y = \epsilon_1 \rho_{xx} + \epsilon_2 \rho_{yy} + F_{mass}(x, y), \tag{3.6}$$

where $F_{mass}(x, y)$ is the new source term for mass. To obtain the new source terms, manufactured analytical solutions ρ, u, v from equation(3.5) are inserted into equation (3.6) while neglecting the time derivative. To prove the convergence rate for spatial discretization of the PDEs via MMS procedure, the analytical equations for the new source terms can be obtained from

$$F_{mass}(x, y) = (\rho u)_x + (\rho v)_y - \epsilon_1 \rho_{xx} - \epsilon_2 \rho_{yy}, \tag{3.7}$$

and they are

$$\begin{aligned} F_{mass}(x, y) = {} & 2\pi cos(2\pi x) sin(\pi x) cos(\pi y) sin(3\pi y) - 3\pi sin(2\pi x) cos(3\pi y) cos(\pi x) sin(\pi y) \\ & + (4\epsilon_1 + 9\epsilon_2)\pi^2 sin(2\pi x) sin(3\pi y). \end{aligned}$$

Similarly, the source terms for the momentum equations in the x and y directions can be determined from the conservation of momentum equations

$$\cancelto{0}{(\rho u)_t} + (\rho u^2 + p)_x + (\rho uv)_y = \epsilon_1(\rho u)_{xx} + \epsilon_2(\rho u)_{yy} + F_{x\text{-}moemntum}(x, y),$$

where $F_{x\text{-}momentum}$ can be written as

$$F_{x\text{-}momentum}(x, y) = (\rho u^2 + p)_x + (\rho uv)_y - \epsilon_1(\rho u)_{xx} - \epsilon_2(\rho u)_{yy}. \tag{3.8}$$

The analytical form of the source terms for the momentum in y direction is derived from the following equation

$$\cancelto{0}{(\rho v)_t} + (\rho uv)_x + (\rho v^2 + p)_y = \epsilon_1(\rho v)_{xx} + \epsilon_2(\rho v)_{yy} + F_{y\text{-}moentum}(x, y),$$

then $F_{y\text{-}momentum}(x, y)$ becomes

$$F_{y\text{-}momentum}(x, y) = (\rho uv)_x + (\rho v^2 + p)_y - \epsilon_1(\rho v)_{xx} - \epsilon_2(\rho v)_{yy}. \tag{3.9}$$

Source terms for the momentum equations in the x and y directions can then be found explicitly by inserting the manufactured solutions (equation (3.5)) into equations (3.8) and (3.9). The final form of the source terms is:

$$\begin{aligned}
F_{x\text{-}momentum}(x, y) = {} & 2\pi sin(3\pi y)sin(\pi x)cos^2(\pi y)(cos(2\pi x)sin(\pi x) + sin(2\pi x)cos(\pi x)) \\
& + 4\pi sin(2\pi x)cos(2\pi x)sin^3(3\pi y) \\
& + \epsilon_1\pi^2 sin(3\pi y)cos(\pi y)(-5sin(2\pi x)sin(\pi x) + 4cos(2\pi x)cos(\pi x)) \\
& + \pi sin(2\pi x)sin(\pi x)cos(\pi x)(-3cos(3\pi y)cos(\pi y)sin(\pi y) + sin(3\pi y)sin^2(\pi y) \\
& - sin(3\pi y)cos^2(\pi y)) \\
& + \epsilon_2\pi^2 sin(2\pi x)sin(\pi x)(10sin(3\pi y)cos(\pi y) + 6cos(2\pi y)sin(\pi y))
\end{aligned}$$

and

$$
\begin{aligned}
F_{y\text{-}momentum}(x,y) = {} & \pi sin(3\pi y)cos(\pi y)sin(\pi y)(2cos(2\pi x)sin(\pi x)cos(\pi x) - sin(2\pi x)cos^2(\pi x) \\
& + sin(2\pi x)sin^2(\pi x) + 2sin(2\pi x)cos^2(\pi x)) \\
& + \pi sin(2\pi x)cos(3\pi y)(3cos^2(\pi x)sin^2(\pi y) + 9sin(2\pi x)sin^2(3\pi y)) \\
& - \epsilon_1\pi^2 sin(3\pi y)sin(\pi y)(5sin(2\pi x)cos(\pi x) + 4cos(2\pi x)sin(\pi x)) \\
& - \epsilon_2\pi^2 sin(2\pi x)cos(\pi x)(10sin(3\pi y)sin(\pi y) - 6cos(3\pi y)cos(\pi y)).
\end{aligned}
$$

Finally, differentiation of the initial analytical manufactured solutions (3.5) are substituted into the conservation of energy equation to determine the source term of the energy equation. The conservation of energy equation is then

$$
\cancelto{0}{E_t} + (u(E+p))_x + (v(E+p))_y = \epsilon_1 E_{xx} + \epsilon_2 E_{yy} + F_{energy}(x,y)
$$

and F_{energy} is

$$
F_{energy}(x,y) = (u(E+p))_x + (v(E+p))_y - \epsilon_1 E_{xx} - \epsilon_2 E_{yy}. \tag{3.10}
$$

The explicit form of (3.10) is

$$
\begin{aligned}
F_{energy}(x,y) = {} & \pi cos(\pi x)cos(\pi y)sin(2\pi x)sin(3\pi y)(sin^2(\pi x)cos^2(\pi y) + cos^2(\pi x)sin^2(\pi y)) \\
& + \pi sin(\pi x)cos(\pi y)cos(2\pi x)sin(3\pi y)(sin^2(\pi x)cos^2(\pi y) + cos^2(\pi x)sin^2(\pi y)) \\
& - 1.5\pi cos(\pi x)sin(\pi y)sin(2\pi x)cos(3\pi y)(sin^2(\pi x)cos^2(\pi y) + cos^2(\pi x)sin^2(\pi y)) \\
& + \pi sin(\pi x)cos(\pi y)sin(2\pi x)sin(3\pi y)sin(\pi x)cos(\pi x)(sin(cos^2(\pi y) - sin^2(\pi y)) \\
& + \frac{\gamma}{\gamma-1}\pi cos(\pi y)sin(2\pi x)sin^3(3\pi y)(cos(\pi x)sin(2\pi x) + 4sin(\pi x)cos(2\pi x)) \\
& - \pi cos(\pi x)sin^2(\pi y)sin(2\pi x)sin(3\pi y)cos(\pi y)(-sin^2(\pi x) + cos^2(\pi x)) \\
& - \frac{\gamma}{\gamma-1}\pi cos(\pi x)sin2(2\pi x)sin^2(3\pi y)(cos(\pi y)sin(3\pi y) + 9sin(\pi y)cos(3\pi y)) \\
& + 2\epsilon_1\pi^2 sin(2\pi x)sin(3\pi y)(sin^2(\pi x)cos^2(\pi y) + cos^2(\pi x)sin^2(\pi y)) \\
& - 4\epsilon_1\pi^2 cos(2\pi x)sin(3\pi y)sin(\pi x)cos(\pi x)(cos^2(\pi y) - sin^2(\pi y))
\end{aligned}
$$

$$
\begin{aligned}
&- \epsilon_1\pi^2 sin(2\pi x)sin(3\pi y)((cos^2(\pi x) - sin^2(\pi x))(cos^2(\pi y) - sin^2(\pi y)) \\
&- \frac{8\epsilon_1\pi^2}{\gamma - 1}(cos^2(2\pi x)sin^3(3\pi y) - sin^2(2\pi x)sin^3(3\pi y)) \\
&+ 4.5\epsilon_2\pi^2 sin(2\pi x)sin(3\pi y)(sin^2(\pi x)cos^2(\pi y) + cos^2(\pi x)sin^2(\pi y)) \\
&- 6\epsilon_2\pi^2 sin(2\pi x)cos(3\pi y)cos(\pi y)sin(\pi y)(cos^2(\pi x) - sin^2(\pi x)) \\
&\epsilon_2\pi^2 sin(2\pi x)sin(3\pi y)((cos^2(\pi x) - sin^2(\pi x))(cos^2(\pi y) - sin^2(\pi y))) \\
&- \frac{27\epsilon_2\pi^2}{\gamma - 1}(2sin^2(2\pi x)sin(3\pi y)cos^2(3\pi y) - sin^2(2\pi x)sin^3(3\pi y)).
\end{aligned}
$$

3.1.4 Verification of Designed Computer Code : Convergence Rate

A convergence rate test is an extremely powerful tool for proving code verification. It is a common practice to compare the formal order of accuracy and the observed order of accuracy for the verification of a designed code [83]. There must be agreement between these two different convergence rates. Formal order of accuracy is defined from the power of mesh sizes Δx or Δt in the leading terms of the truncation error, obtained by applying a numerical scheme (finite difference, finite volume, finite element, etc.) to the governing PDEs. On the other hand, computation of the observed order of accuracy is based on discretization error which based on the difference between the analytical solution (benchmark solution) of the governing partial differential equation and the approximate solution of the discrete equation via a discrete error norm. To calculate the observed order of accuracy, the global numerical error should be monitored as the mesh size goes to zero, $\Delta x \rightarrow 0$. This error is calculated via the L^2 norm of the discretization error

$$
L^2\ norm = \left(\Delta x \Delta y \sum_{i=1}^{M} \sum_{j=1}^{N} (u_{i,j}^{exact} - u_{i,j}^{approximate})^2 \right)^{\frac{1}{2}} \tag{3.11}
$$

where $(u_{i,j}^{exact} - u_{i,j}^{approximate})$ is the difference between the exact and approximate solutions. Here, $u_{i,j}^{exact}$ is the analytical solution and $u_{i,j}^{approximate}$ is the corresponding numerical solution. The benchmark solution plays a crucial role in code verification and is generated by using the MMS method. The convergence rate is computed by

using equation (3.11) as

$$q = log_{10}\left(\frac{L^2\ norm_{k_1}}{L^2\ norm_{k_2}}\right) \Big/ log_{10}\left(\frac{k_1}{k_2}\right) \tag{3.12}$$

where k_1 and k_2 refer to fine and coarse mesh sizes, respectively. Once an agreement is obtained between the formal and observed convergence rates, then the current numerical code can be verified from the MMS procedure. In the following section the implementation of all of these procedures into the designed code will be discussed. Furthermore in the next section, the discretization procedure for the spatial discretization and results will be presented to illustrate the robust performance of the proposed methods.

3.2 Spatial Discretization Numerical Method

The chosen numerical method to discretize the governing PDE is the second order cell centered finite volume scheme [50]. In this part of the study, spatial discretization is the only concern, so the elaboration of this numerical scheme is presented for only flux derivations of the space variables x and y, $F(u(x,y))_x$ and $G(u(x,y))_y$. Then, it is necessary to present the discretized form of the steady state system of the conservation laws as follows,

$$\frac{F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}}{\Delta x} + \frac{G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}}}{\Delta y} = 0, \tag{3.13}$$

where $F_{i\pm\frac{1}{2},j}$ and $G_{i,j\pm\frac{1}{2}}$ are the numerical flux functions of cell average physical fluxes

$$\begin{aligned} F_{i\pm1/2,j} &\approx \tfrac{1}{\Delta y}\textstyle\int_{y_{j-1/2}}^{y_{i+1/2}} F(u(x_{i\pm1/2}, y))dy \\ G_{i,j\pm1/2} &\approx \tfrac{1}{\Delta x}\textstyle\int_{x_{i-1/2}}^{x_{i+1/2}} G(u(x, y_{j\pm1/2}))dx. \end{aligned} \tag{3.14}$$

Here, $x_{i+1/2} = (x_i + x_{i+1})/2$ and $y_{j+1/2} = (y_j + y_{j+1})/2$ are the mid points, and $\Delta x = (x_{i+1/2} - x_{i-1/2})$ and $\Delta y = (y_{j+1/2} - y_{j-1/2})$ are the mesh sizes in the x and y directions, respectively. Note that since we work with uniform mesh sizes, the values of Δx and Δy always remain same constants. The modified discrete fluxes with numerical

diffusion are also presented as

$$
\begin{aligned}
F_{i\pm\frac{1}{2},j} &= \frac{F_{i\pm1,j}+F_{i,j}}{2} \mp \epsilon_1 \left(\frac{u_{i\pm1,j}-u_{i,j}}{\Delta x}\right) + O(\Delta x^2) \\
G_{i,j\pm\frac{1}{2}} &= \frac{G_{i,j\pm1}+G_{i,j}}{2} \mp \epsilon_2 \left(\frac{u_{i,j\pm1}-u_{i,j}}{\Delta y}\right) + O(\Delta y^2),
\end{aligned}
\tag{3.15}
$$

where $O(\Delta x^2)$ and $O(\Delta y^2)$ are the truncation errors from Taylor series expansion, and $\epsilon_{1,2}$ are the coefficients of the artificial diffusion for F and G fluxes, respectively. The discretization procedure is also constructed to satisfy consistency between steady state system of conservation laws

$$F(u(x,y))_x + G(u(x,y))_y = 0,$$

and corresponding numerical flux derivations in equation (3.15) as $\Delta x \to 0$, $\Delta y \to 0$. Here, $F(u(x,y))$ and $G(u(x,y))$ are smooth flux functions for the conservative variables, $\mathbf{u}(x,y) = (u_1(x,y), \ldots u_N(x,y))^T$.

3.3 Numerical Simulations and Results

In this chapter, the verification of the designed code is presented. Since the non-linear Euler equations of gas dynamics do not possess exact solutions in general, the MMS procedure was followed for this purpose to verify the written numerical code. Here a comparison between the observed and the formal convergence rates is conducted only for spatial convergence to prove the accuracy of the designed code, since the temporal convergence rate is not addressed in this section. The manufactured solutions presented in equation (3.5) with the modified source terms are substituted into the discretized governing equations as stated in equation (3.13) and the corresponding numerical solutions are obtained. The source terms for each flux variables, density, momentum in x and y directions and energy, are derived from equations (3.7),(3.8),(3.9) and (3.10). The advantage of using the MMS procedure is that there is no need to concern about the computational domain [76]. Any domain which is convenient for solving the governing equations can be chosen. Here, the chosen computational domains

for the current numerical calculations are $[0,1] \times [0,1]$ in x, y coordinates, respectively.

Solutions for the interior part of the numerical scheme are obtained from numerical simulation. Since the argument for verification of the computer code is based on comparison between the approximate solution and the analytical manufactured solutions, the agreement between the convergence rates of the observed and formal ones have to be shown. Thus, the discretization of $L^2 error$ norm from equation (3.11) plays a crucial role in comparing the analytical and numerical results. Once the desired results are obtained, the observed order of accuracy can be computed by refining the mesh grid and comparing the results of two consecutive discrete mesh levels with equation (3.12). In Table 3.1, the outputs from numerical simulations for $L^2 error$ norm and convergence rates of each flux variables are summarized for four different mesh sizes, $\Delta x = \Delta y = 0.1,\ 0.05,\ 0.025,\ 0.0125$. The results of these calculations indicate that, the L^2 norm of errors decrease almost by a factor of four as the mesh sizes are refined, while the observed order of accuracies converge to the expected formal convergence rate of two. The obtained results support the accuracy of the proposed numerical method to approximate the solution with the correct order of accuracy.

Table 3.1: L^2 error norm and convergence rate for second order finite volume method

M	N	$L^2(\rho)$	q_ρ	$L^2(E)$	q_E	$L^2(\rho v)$	$q_{\rho v}$	$L^2(\rho u)$	$q_{\rho u}$
10	10	0.0046	-	0.0647	-	0.0028	-	0.6418	-
20	20	0.0013	1.8270	0.0173	1.9071	0.0009	1.6330	0.1829	1.8113
40	40	0.0003	1.9677	0.0047	1.8833	0.0002	1.8834	0.0470	1.9610
80	80	0.0001	1.9934	0.0012	1.9717	0.0001	1.9515	0.0118	1.9905

The results presented in Table 3.1 are depicted graphically in Figure 3.1. In this figure, the L^2 error norm for each conserved variable is given as a function of four different spatial mesh sizes on a log-log scale. These are compared with the slope for second order accurate numerical method. The L^2 error decreases with mesh size, as required. It can be seen that the error for each variable decreases with mesh size in accord with second order accuracy.

In Figure 3.2, the manufactured analytical solutions for the density, x and y momentum and energy for the compressible Euler equations of gas dynamics (see equation 3.5) are presented. These specific solutions allow us to obtain numerical results for the

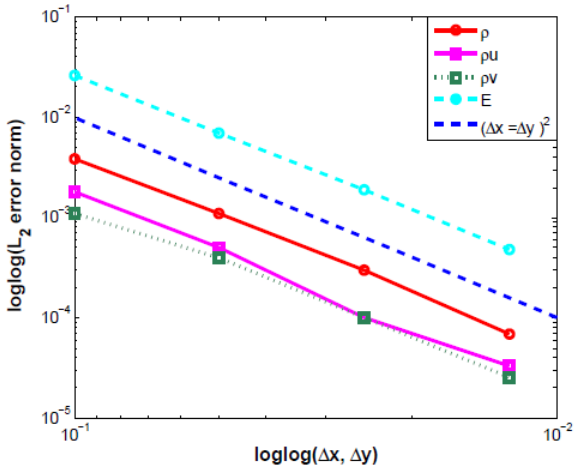


Figure 3.1: The dark blue (dashed) line denoted by $(\Delta x = \Delta y)^2$, demonstrates the theoretical error variation for a numerical method of second order of accuracy.

entire computational domain, even if a coarse mesh is used. Then, after implementing these benchmark solutions, including the new source terms, into the numerical method, the error is calculated by comparing the analytical and numerical solutions, i.e., $\left(u_{i,j}^{exact} - u_{i,j}^{approximate}\right)$. In Figure 3.3, the error in the density for each mesh size is illustrated. It can be observed that the error is decreasing as the mesh size decreases, as required. Also, in Figures 3.4, 3.5 and 3.6, comparisons between the benchmark and numerical solutions are illustrated for the momentum in the x and y directions and the energy, respectively. These results were used to calculate the L^2 error norm (see equation (3.11)). It can be seen that the errors are $(\Delta x)^2$, as expected.

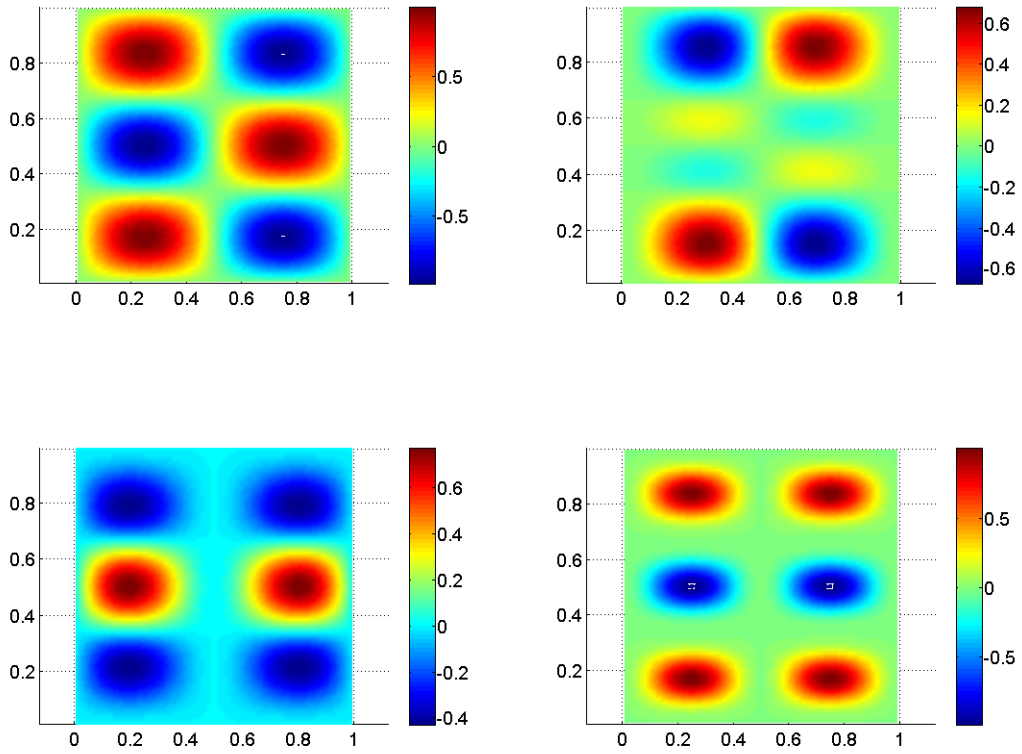


Figure 3.2: 2D MMS analytical solutions; density function ρ(at the top left), momentum function in -x direction ρu(at the top right), momentum function in -y direction, ρv(at the bottom left), energy function E(at the bottom right)

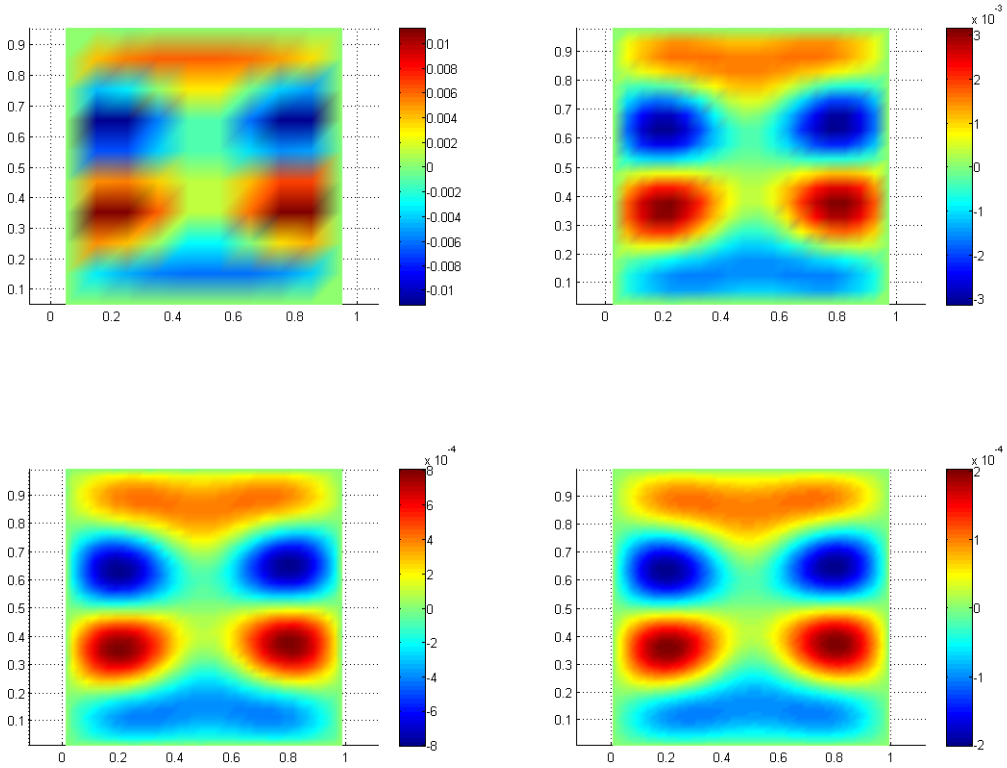


Figure 3.3: Errors between exact and numerical solutions of density, ρ, according to different mesh sizes, $\Delta x = 0.1$, $\Delta y = 0.1$(at top left), $\Delta x = 0.05$, $\Delta y = 0.05$(at top right), $\Delta x = 0.025$, $\Delta y = 0.025$(at bottom left), $\Delta x = 0.0125$, $\Delta y = 0.0125$(at bottom right)

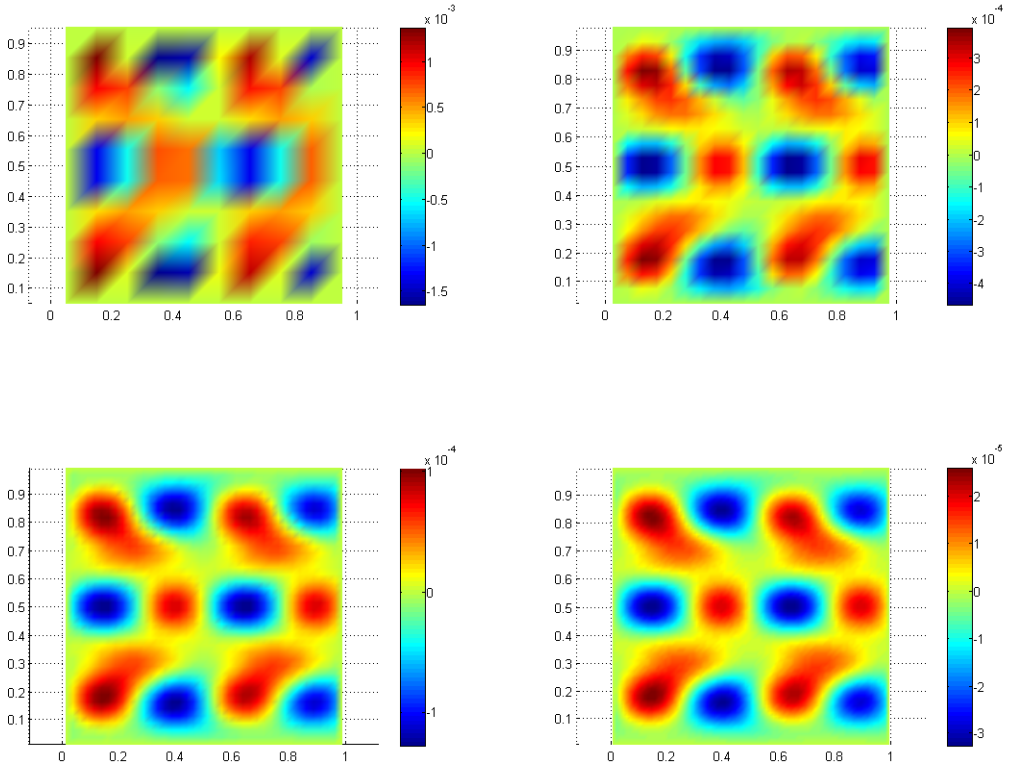


Figure 3.4: Errors between exact and numerical solutions of momentum in -x direction, ρu, according to different mesh sizes, $\Delta x = 0.1$, $\Delta y = 0.1$(at top left), $\Delta x = 0.05$, $\Delta y = 0.05$(at top right), $\Delta x = 0.025$, $\Delta y = 0.025$(at bottom left), $\Delta x = 0.0125$, $\Delta y = 0.0125$(at bottom right)

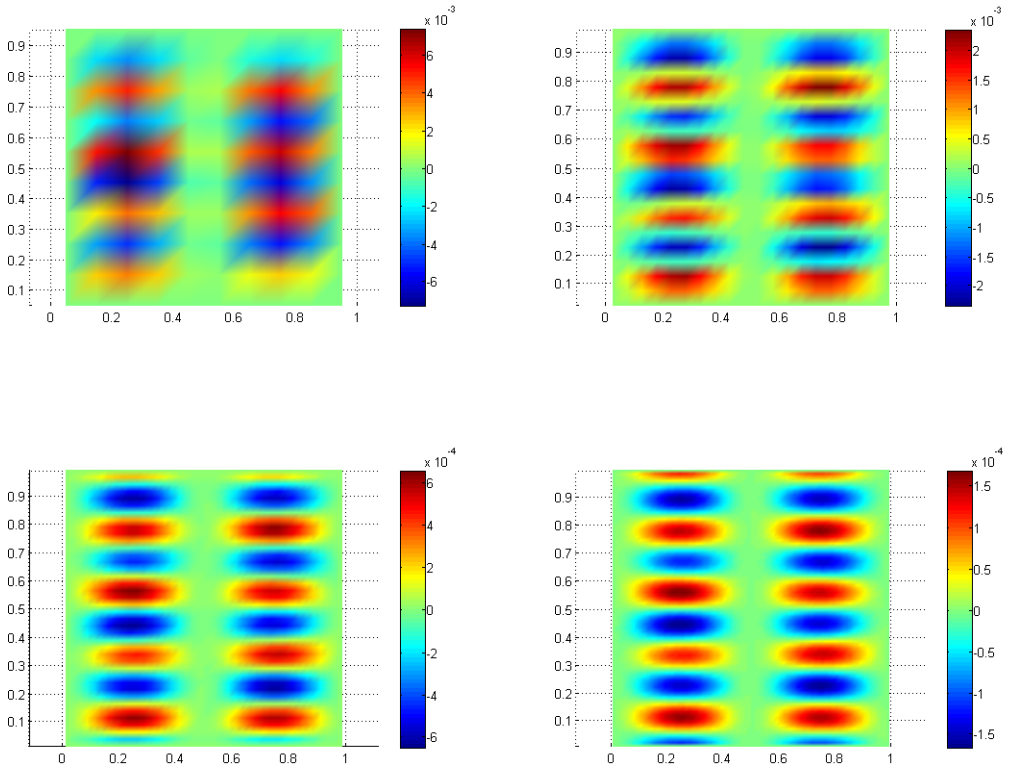


Figure 3.5: Errors between exact and numerical solutions of momentum in -y direction, *rhov*, according to different mesh sizes, $\Delta x = 0.1$, $\Delta y = 0.1$(at top left), $\Delta x = 0.05$, $\Delta y = 0.05$(at top right), $\Delta x = 0.025$, $\Delta y = 0.025$(at bottom left), $\Delta x = 0.0125$, $\Delta y = 0.0125$(at bottom right)

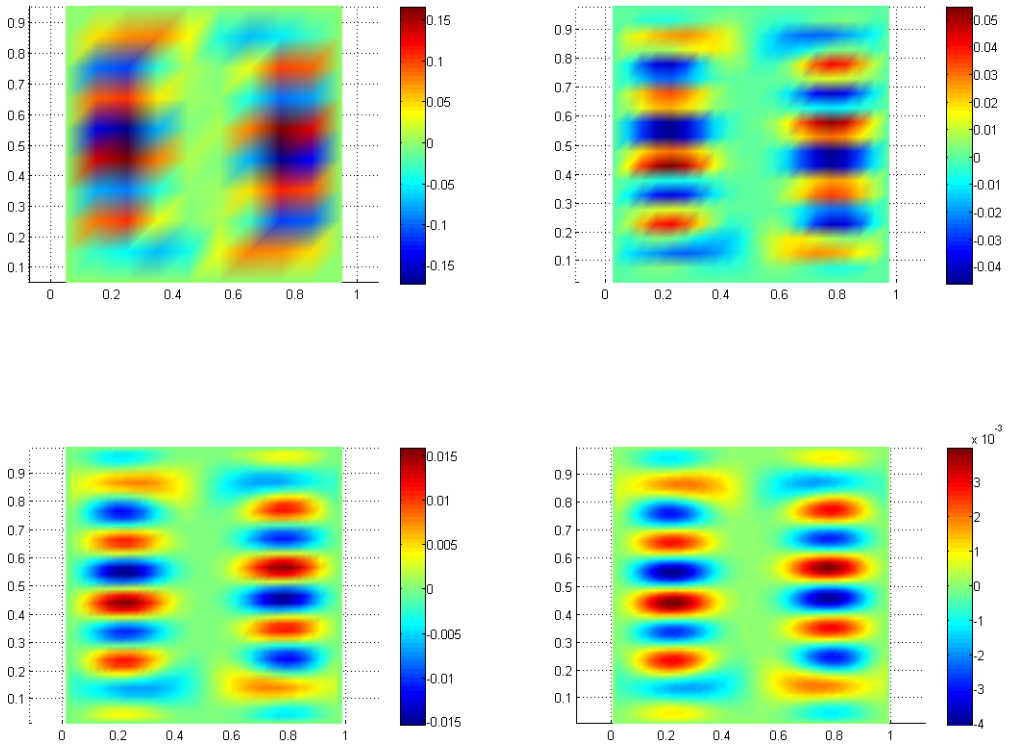


Figure 3.6: Errors between exact and numerical solutions of energy, E, according to different mesh sizes, $\Delta x = 0.1$, $\Delta y = 0.1$(at top left), $\Delta x = 0.05$, $\Delta y = 0.05$(at top right), $\Delta x = 0.025$, $\Delta y = 0.025$(at bottom left), $\Delta x = 0.0125$, $\Delta y = 0.0125$(at bottom right)

Chapter 4

The Sod's Shock Tube Problem

In this section, the validation of the developed numerical code for the two-dimensional non-linear hyperbolic system of conservation laws is presented. In particular, our main interest is to obtain numerical solutions for the Euler equations of compressible gas dynamics. However, since such non-linear equations can contain discontinuities (shocks) in their solutions, there are limited analytical solutions to test the developed numerical algorithm. One of the strategies to test the developed code for two-dimensional problems is to solve a one dimensional test case for x and y directions, independently. In Sod [54], a simple one dimensional shock tube problem was introduced as a common test problem for inviscid, compressible fluid dynamic problems. Although the test case proposed by Sod [54] can be easily implemented into numerical schemes, it is very challenging for any numerical algorithm to capture all the physical features, such as shock waves, rarefaction waves and contact discontinuities. This test case reveals how well the developed numerical method approaches the exact solution, and it has been applied to real world applications, such as the development of supersonic aircraft, improvement of gun performance, the investigation of asteroid impacts and shuttle atmospheric entry [84]. We shall now give a physical description of the Sod's Shock Tube Problem.

4.1 Introduction

The study given by Sod [54] was based on a long one-dimensional shock tube for which the schematic diaphragm is given in Figure 4.1. A thin diaphragm was positioned in the middle of the tube with the gas pumped to high pressure on one side of the

diaphragm. The regions, on the right and left side of the diaphragm, can be filled with the same or different gases. In our study, the two regions are filled with the same gas, which is assumed to be air. The pressure and density of the gas are different in the right and left regions. Here, the initial values of the pressure and density in SI units were taken from Hirsch [18]. These values are for air as a perfect gas with a specific heat ratio γ of 1.4. The initial contribution is

$$\rho(x,0) = \begin{cases} 1.0\ kg/m^3 & if\ \ x < M/2, \\ 0.125\ kg/m^3 & if\ \ x \geq M/2 \end{cases} \tag{4.1}$$

$$p(x,0) = \begin{cases} 10^5\ Pa & if\ \ x < M/2, \\ 10^4\ Pa & if\ \ x \geq M/2 \end{cases} \tag{4.2}$$

with the gas at rest

$$u(x,0) = 0\ m/s. \tag{4.3}$$

Here M is the length of the tube. As presented in Figure 4.1, the gas is at rest with zero velocity at $t = 0$ in the both regions. The diaphragm is then instantaneously broken or removed. This generates the flow. While the system is trying to reach a balance between high and low pressure gases, the initial two regions transform into five different regions as in Figure 4.1. Among these five regions, the left high pressure

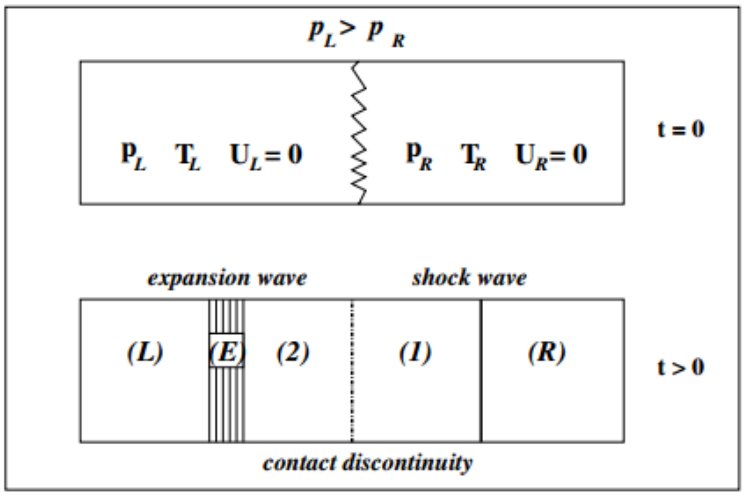


Figure 4.1: Sketch of the regions initially at $t = 0$ and after breaking the diaphragm at $t > 0$. This figure is taken from Danaila et al. [84].

region (L) and the right low pressure region (R) are the regions undisturbed from the initial state. The remaining three regions are due to the propagation of the waves which are caused as a result of the difference in initial pressure at $t = 0$. The region (E) contains an isentropic expansion wave which moves to the left and expands in time. Furthermore, the shock wave, which is caused by compression of low pressure gas and is located between the regions (1) and (R), moves in the opposite direction. A contact discontinuity is located between the regions (1) and (2), and separates the shock and expansion waves. It also moves to the same direction as the shock wave. We then need to calculate this flow and separates the gas initially on either side of the diaphragm. While the parameters are continuous in the region (E), singularities occur in the other regions. For instance, while the pressure and the velocity parameters are continuous and equal in the regions (1) and (2), $p_1 = p_2$ and $u_1 = u_2$, a discontinuity exists for the density and momentum across the contact discontinuity. There is also a general jump for all flow parameters, i.e., density, pressure, energy and velocity across the shock wave. In the following sections, a numerical method is implemented to solve the Euler equations using the test case proposed by Sod [54], and the results are displayed to verify the accuracy of the numerical code.

4.2 Governing Equations

The development of an efficient and robust numerical code is required to solve the non-linear Euler equations of gas dynamics in 2D. To ensure the performance of this numerical code, the governing equations are tested using the one-dimensional Sod's shock tube test case. The one dimensional governing equation is initially tested for one direction and then the domain is turned around 90° to test for the other direction. This procedure is followed to be sure the code does not have any errors in either the x and y directions.

The following procedure has been followed for this purpose. First, flow is assumed to be in the x direction by taking the velocity in the y direction to be zero, $v = 0$. The gas equations are then of the form

$$\mathbf{u}_t + f(\mathbf{u})_x = 0, \tag{4.4}$$

where $\mathbf{u}$ is the unknown solution vector and $f(\mathbf{u})$ is the corresponding flux vector in x direction. The matrix form of the unknown solution and corresponding flux vectors can be stated as follows:

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}, \quad f(\mathbf{u}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{bmatrix}. \tag{4.5}$$

Here, ρ is the density, u is the velocity in x direction, E is the total energy and p is the pressure, which can be written from the equation of state for a perfect gas as [15]

$$p = (\gamma - 1)\left(E - \frac{\rho u^2}{2}\right),$$

where γ is the ratio of specific heats. In the next step, the one dimensional Euler equations in y direction can be stated by taking the velocity in the x direction to be zero, $u = 0$. The velocity u in equation (4.5) is replaced by the velocity v in the y direction. After stating the PDEs from a physical perspective, the discretization procedure can be defined by using the cell-centered finite volume (CCFVM) and Strong Stability Preserving third order Runge-Kutta (SSPRK3) methods for space and time discretizations, respectively [49, 85]. These methods have been used to find numerical solutions which converge to the exact solution of the Euler equations of gas dynamics. The explanations of the implemented methods for spatial discretization and time integration are presented in Sections 4.2.1 and 4.2.2, respectively.

4.2.1 Spatial Discretization

In this part, the CCFVM is used for the spatial discretization process [85]. Since the CCFVM is designed for the integral form of conservation equations, it is perfectly convenient to apply this method to governing equations being a canonical example of conservation laws [50]. In CCFVM, the computational domain is divided into many sub-domains, which are called control volumes. Since only a one-dimensional system is considered here, the computational spatial domain is divided into sub finite cells instead,

$$D_i = [x_{i-1/2}, x_{i+1/2}],$$

where $x_{i\pm1/2} = (x_i + x_{i\pm1})/2$ are the mid-points. The approximated integral forms of the exact solution variables, u, over each sub finite cell, D_i, are interpreted as

$$u_i \approx \frac{1}{\Delta x} \int_{D_i} u(x,t)dx, \tag{4.6}$$

where $\Delta x = x_{i+1/2} - x_{i-1/2}$ is the spatial mesh length. Here, u_i is the approximate value of u in the ith cell. The integral form of the governing Euler equations can be written over each sub finite cells as

$$\frac{\partial}{\partial t} \int_{D_i} u(x,t)\ dx = f(u(x_{i-1/2},t)) - f(u(x_{i+1/2},t)) \tag{4.7}$$

where the solution vector u and corresponding flux vector f are given in equation (4.5). It is a very challenging task to evaluate nonlinear equations in integral form since the exact solution does not generally exist. As a result, the integrated PDEs are expressed as discretized algebraic equations as,

$$(u_i)_t + \frac{f_{i+1/2} - f_{i-1/2}}{\Delta x} = 0 \quad i = 1, \ldots, N. \tag{4.8}$$

There are two important points that need to be considered for the numerical fluxes. First, the consistency between numerical and differential fluxes are crucial for the system, i.e., $f_{i+1/2} \approx f(u(x_{i+1/2},t))$. This is because that approximation of the exact solution is achieved with the numerical solution calculated from the numerical method. Second, the flux f needs to be modified by adding artificial diffusion to suppress possible nonphysical oscillations that are created by the discretized PDE in the region of shocks. Hence, the numerical fluxes in equation (4.8) are modified to

$$(u_i)_t + \frac{f^\star_{i+1/2} - f^\star_{i-1/2}}{\Delta x} = 0, \tag{4.9}$$

where

$$f^\star_{i+1/2} = \frac{f_{i+1} + f_i}{2} - \epsilon_1 \frac{u_{i+1} - u_i}{\Delta x}. \tag{4.10}$$

Here, $f^\star_{i\pm1/2}$ refers to the modified flux functions and ϵ_1 is a small positive artificial diffusion coefficient. In this scheme, while the unknown flow variables are evaluated

at the middle of the sub finite cells in the computational domain, the flux variables are calculated by taking the average of the flow variables from the two neighboring sub finite cells and are positioned at the edges of these cells. They basically provide information about the quantity of gas flow that goes in and out of the neighboring cells.

4.2.2 Temporal Integration

The semi discretized numerical method is now implemented into the governing PDEs in this section. This discretization process is commenced with the stable CCFVM spatial discretization method of the previous section. Then the system turns into the form of a semi-discrete ordinary differential equation (ODE). Here, a strong stability preserving Runge-Kutta scheme for time discretization is applied to approximate the PDEs [49]. This time discretization method is also called a total variation diminishing Runge-Kutta scheme, for which further details can be found in [86]. SSPRK schemes result in stable results from time discretization by preventing any possible growth in the solution in time and also improve the time step limitation of the CFL (Courant, Friedrichs, and Lewy) condition restriction. They are employed to obtain non-oscillatory results particularly when solving non-linear hyperbolic equations. In Gottlieb et al. [87], the robustness of this method was proven by comparing numerical simulations obtained using SSP and non-SSP time discretizations. It was found that the non-SSP time discretization methods cannot counteract oscillations existing in the numerical solution even if a stable spatial discretization is chosen [87]. Therefore, it is more appropriate to use SSP methods for the time discretization to approximate the solution of the PDEs. In SSP methods, all coefficients of the general Runge-Kutta schemes are set to be nonnegative as in [87],

$$\begin{aligned} u^{(0)} &= u^n, \\ u^{(i)} &= \sum_{k=1}^{i-1} \left(\alpha_{ik} u^{(k)} + \Delta t \beta_{ik} L\left(u^{(k)}\right) \right), \quad i = 1, \ldots, m \\ u^{(n+1)} &= u^{(m)}, \end{aligned} \tag{4.11}$$

where $\alpha_{ik} \geq 0$ and $\beta_{ik} \geq 0$. At this stage, the third order SSPRK method can be introduced for time discretization for the gas equations. This method is designed to

solve ODEs of the form

$$\frac{d}{dt}\mathbf{u} = L(\mathbf{u}),$$

where L is a spatial operator and the SSPRK3 is given as

$$\begin{aligned} u^{(1)} &= u^n + \Delta t L(u^n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^{(1)}), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^{(2)}). \end{aligned} \tag{4.12}$$

Here it is assumed that the CFL coefficient is equal to one, CFL= 1 [87].

4.2.3 CFL condition

The CFL condition also known as Courant number in the literature and is defined as

Lemma 1. *A numerical method can be convergent only if its numerical domain of dependence contains the true domain of dependence of the PDE, at least in the limit as Δt and Δx go to zero [50].*

This is a fundamental necessary condition, but it is not sufficient to ensure the stability of any explicit numerical schemes, i.e., finite difference and finite volume methods [50]. There are two different domains of dependence which are the physical and the numerical domains of dependence. To obtain numerical solutions which converge towards the solution of the partial differential equation, the numerical domain of dependence must include that of the physical domain as the grid is refined (resolution is increased), i.e., $\Delta x \to 0$, $\Delta t \to 0$, as sketched in Figure 4.2.

The reason is that the information of the solution propagates at a finite speed in hyperbolic PDEs [50]. To allow this information to propagate at the correct physical speed during spatial discretization, the time step must be kept small enough. Thus, this condition provides a link between the space and time steps to prevent incorrect numerical solutions. The importance of satisfying this condition is that if the initial data is perturbed, not only the exact solution, but also the numerical solution varies at the same time in order to preserve stability. To this end, the following CFL condition presents the relationship between the time step, the space step and the physical speed

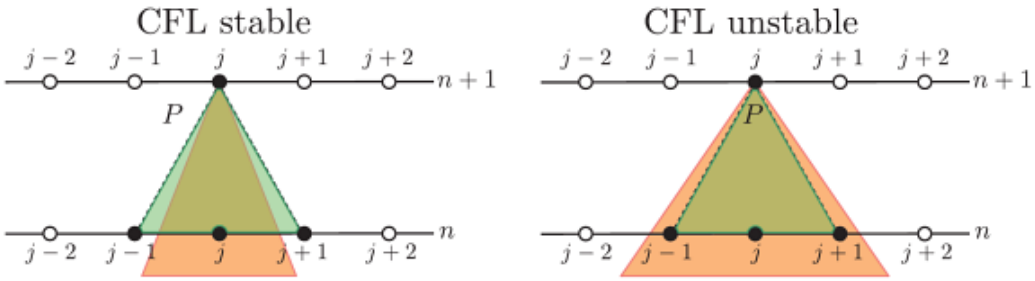


Figure 4.2: Left part is CFL stable since numerical domain of dependence includes physical domain of dependence, $D_{\mathbf{P}} \subset D_{\mathbf{N}}$. However, right part is unstable since this condition is contrary to CFL stability condition, $D_{\mathbf{P}} \not\subset D_{\mathbf{N}}$. Here $D_{\mathbf{P}}$ is physical domain (orange area), $D_{\mathbf{N}}$ is numerical domain (green area). This figure is taken from Rezzolla et al. [88].

of information

$$CFL = \frac{\Delta t}{\Delta x} \max_i \left|\lambda^i\right|, \tag{4.13}$$

where λ^i is one of the eigenvalues of the Jacobian flux, $f(\mathbf{u})_{\mathbf{u}}$, in the system of hyperbolic PDEs. The eigenvalues of the one dimensional Euler equations are

$$\Lambda \;=\; \text{diag}\,[u-a, u, u+a],$$

where Λ is the diagonal matrix with the eigenvalues on the diagonal. The maximum eigenvalue of the Euler equation is $u+a$, where a is the speed of sound, i.e., $a = \sqrt{\frac{\gamma p}{\rho}}$. This maximum eigenvalue determines how fast information propagates during the numerical solution process. Hence, the proper way to capture the physical domain within the numerical domain is to calculate the largest eigenvalue (maximum physical speed) in the governing system of PDEs. That is the reason why the maximum eigenvalue should be selected for the calculation of the time step. Since the numerical scheme of the present work is a centered three point stencil, the CFL stability condition is satisfied when $CFL \leq 1$. Note that by using the three point stencil method, the numerical solution u_i^{n+1} depends on the previously obtained solutions, u_{i-1}^n, u_i^n and u_{i+1}^n.

4.3 Numerical Simulations and Results

In the present section, numerical simulations obtained by using the CCFVM method for space discretization and the SSPRK3 method for time integration are used to vali-

date the numerical code. In the detailed calculations, appropriate boundary conditions for the real problem of the next chapter for a very long tube are used. Since the priority of this test case is not related to boundary issues, the calculations were stopped before the shock wave reaches the right boundary of the tube. The detailed discussion of the boundary issues is given in Chapter 5.

To conduct numerical simulations, the following domain and initial conditions are used. It is assumed that the computational domain is a straight, ten meter long tube with a domain $D_c = [0, 10]$. A diaphragm is located in the middle of the domain at $x_0 = 5\ m$. The initial physical flow parameters are as given in [18, 89] for the left and right regions and are given by (4.1), (4.2), (4.3) for density, pressure and velocity, respectively. The computational domain was divided into 5000 uniform computational cells, with the final time being $T_{end} = 0.0061s$. The CFL condition and artificial diffusion coefficients were set to 1 and 0.4, respectively. The exact solution is calculated by using the shock conditions which are stated in [10] as

$$\begin{aligned} M &= \frac{U - u_1}{a_1} = \left(1 + \frac{\gamma + 1}{2\gamma} z\right)^{1/2} \\ \frac{u_2 - u_1}{a_1} &= \frac{z}{\gamma \left(1 + \frac{\gamma+1}{2\gamma} z\right)^{1/2}} \\ \frac{\rho_2}{\rho_1} &= \frac{1 + \frac{\gamma+1}{2\gamma} z}{1 + \frac{\gamma-1}{2\gamma} z}, \end{aligned}$$

where $z = p_2 - p_1/p_1$ and subscripts $\{_{1,2}\}$ refer to shock strength and flow quantities ahead and behind of the shock waves, respectively.

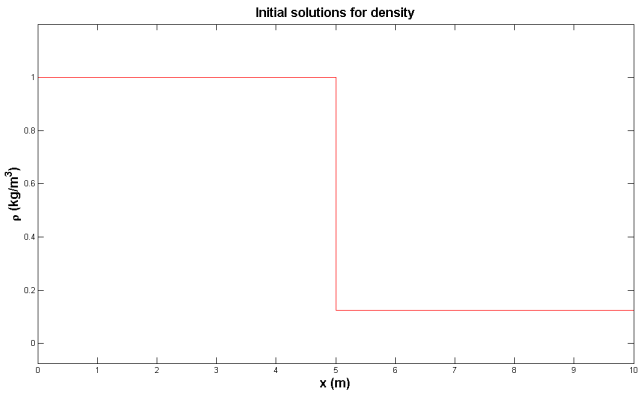
Figure 4.4 presents a comparison between the results of numerical computations and exact solution for the velocity, the pressure, the density, the momentum and the Mach number (locally) as a function of x. It can be seen that there is an excellent agreement between the numerical and exact solutions. First, we consider the density. The initial jump discontinuity for density as in Figure 4.3a moves into two discontinuities, as seen in Figure 4.4a. These are located at the contact discontinuity and the shock wave. The contact discontinuity has been smoothed by the artificial viscosity and is a smooth variation.

The second discontinuity is the shock wave, which is moving to the right and is

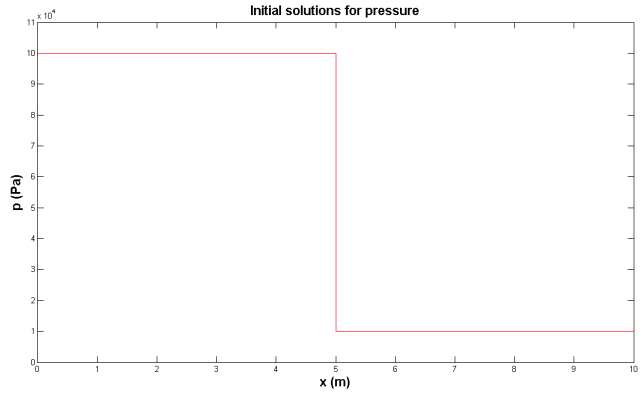
located at $x = 8.3585\ m$ in Figure 4.4a. The expansion wave is in the region between $x = 2.7321\ m$ and $x = 4.8653\ m$. The remaining left and right regions filled with the gas are at their initial rest states. These are in between $x = 0\ m$ and $x = 2.7321\ m$, and $x = 8.3585\ m$ and $x = 10\ m$, respectively. Since the effect of the gas flow has not reached these regions, they are undisturbed. The numerical solution does not have the sharp discontinuities of the exact solution due to the effect of artificial viscosity. In addition, the similar discontinuities are appeared at the same places for momentum variables and Mach number as in Figures 4.4b and 4.4c. However, while discontinuities in momentum are captured in a similar manner with density variable, it is not the same case for Mach number. The reason is that both discontinuity solutions are captured quite close to the exact solution for Mach number as in Figure 4.4c.

There is again an excellent agreement between the exact and numerical solutions when the obtained results of physical quantities are compared with reported ones in [18]. The ratio of the pressure of the 2nd region and the right region is $p_2/p_R = 3.031$, and the pressure of the 1st and 2nd regions is $p_2 = p_3 = 30313\ Pa$ as required as the contact discontinuity cannot sustain a pressure jump. The velocity also stays constant in the contact discontinuity region due to the lack of pressure force. Finally, the velocity in the 2nd region is $u_2 = 293\ m/s$, similar to that reported in the literature [18].

It should be noted the same argument between the exact and numerical solution is obtained for a shock tube in the y direction. The results are presented for the x direction only as the solution in the y direction is identical. As presented in Figure 4.4 there is an excellent agreement between the numerical (dotted lines) and exact (solid lines) solutions. This then verifies the accuracy of the scheme used here.



(a) Initial density value, ρ



(b) Initial pressure value, p

Figure 4.3: The initial values of density and pressure at the right and left sides of the diaphragm which is located in the middle of the domain at $x = 5\ m$ at time $t = 0\ s$.

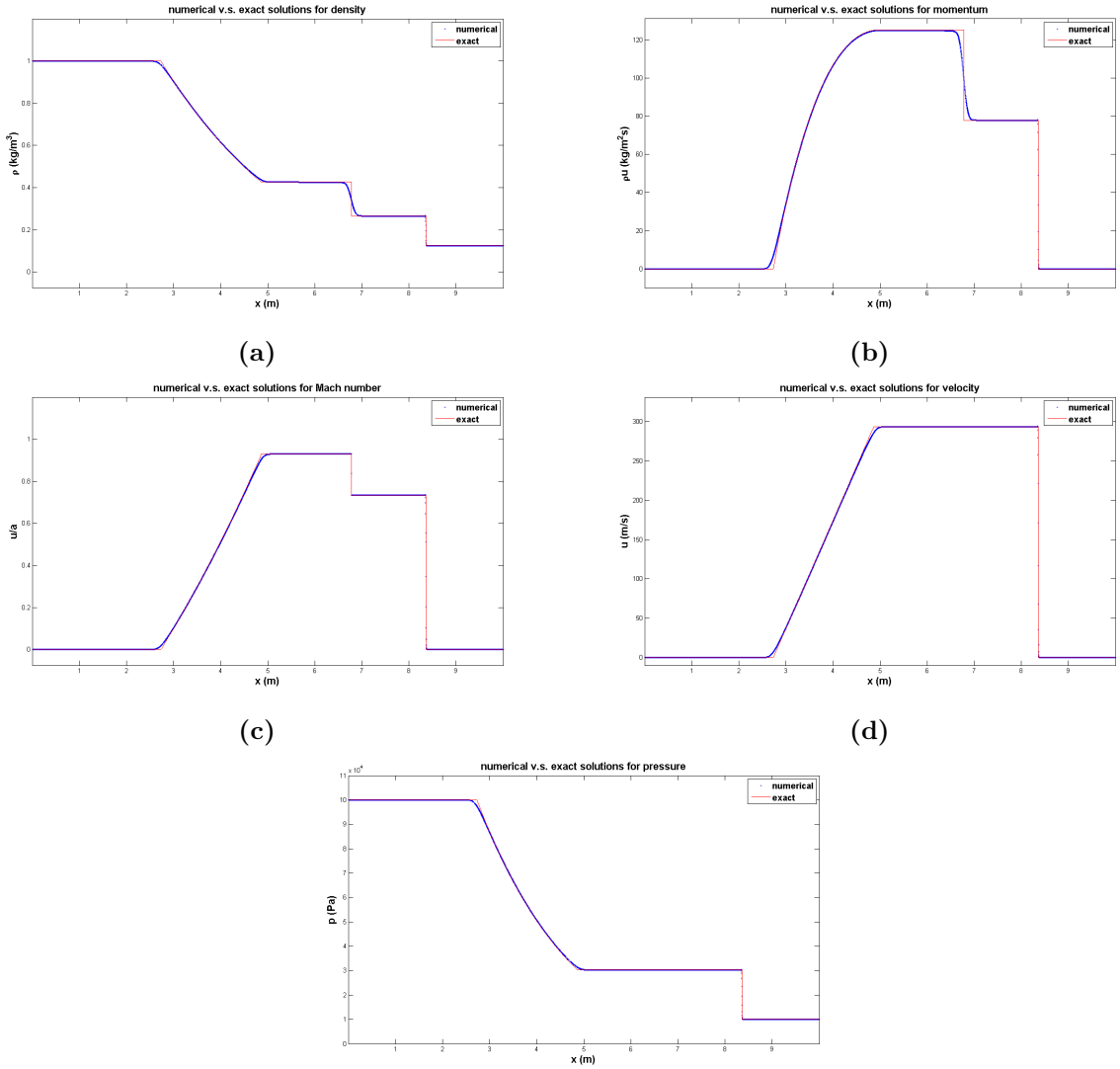


Figure 4.4: Comparison of numerical and exact solutions for a) Density function, ρ b) Momentum function, ρu c) Mach number, u/a, where $a = \sqrt{\frac{\gamma p}{\rho}}$ is speed of sound d) Velocity function, u e) Pressure, p v.s. distance along the wall in the x direction at $t = 0.0061\ s$. They are stated for initial conditions in equations (4.1)-(4.3).

Chapter 5

Shock propagation in curved channels

The aim of numerically studying nonlinear systems of conservation laws is to broaden knowledge of the flow of a compressible gas in non-uniform regions with emphasis on the development of shock and expansion waves. Most of the studies to date have been investigated these physical features based on the propagation of initially planar shock waves in tubes or channels or self-similar flows either numerically or experimentally [4, 10, 42, 43, 44]. However, these studies are challenging due to the nonlinear nature of the equation and the need to capture shock waves. In the present work, the diffraction of shock waves in two-dimensional channels with a 90° bend is studied. The Euler equations of compressible gas dynamics are considered to obtain numerical solutions for the flow field and for the propagation of shock front wave itself. A comparison is carried out between the numerical solutions of the discretized governing compressible gas equations and experimental results reported in the literature [42]. It needs to be noted that if the dimension of the problem is increased from one-dimensional to two-dimensional, finding numerical solutions becomes more challenging. Prior to detailing numerical study, the experimental study [42] is first discussed.

5.1 Experimental Study

In the experimental work of Edwards et al. [42] shock propagation in two different two-dimensional channels with 90° bends, hereinafter called Channel 1 and Channel

2, was considered. The difference between these two channels is due to the different radii of curvature of their quarter circular bends. Figure 5.1 demonstrates the sketch of the length of curvature and channel radii. The inner and outer radii of Channel 1

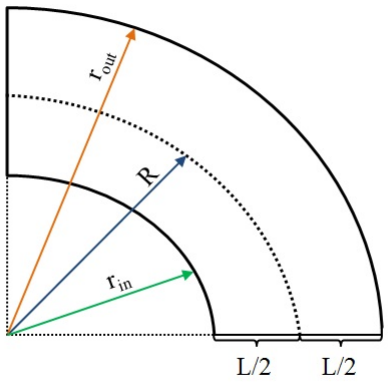
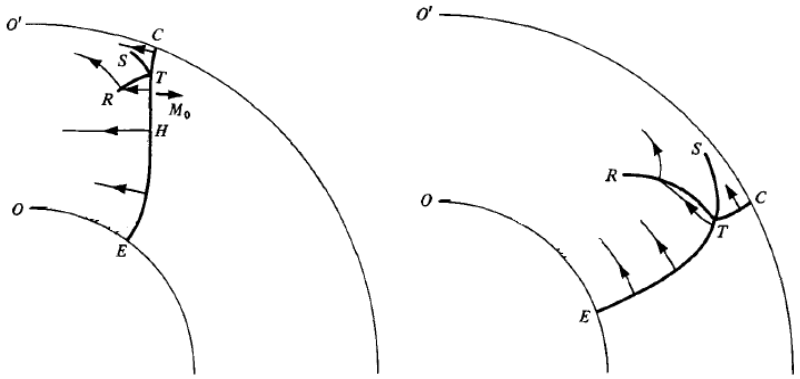


Figure 5.1: Sketch of radii of inner, outer walls and length of curvature for channels. Here L refers to the width of the channel.

are $r_{1,\text{in}} = 48.9mm$ and $r_{1,\text{out}} = 101.1mm$, respectively, and the radius of curvature is $R_1 = 75mm$. For Channel 2, the inner and outer radii are $r_{2,\text{in}} = 123.9mm$ and $r_{2,\text{out}} = 176.1mm$, respectively, and the radius of curvature is $R_2 = 150mm$. When these two channels are compared, the Channel 2 is significantly larger than Channel 1 in terms of the ratio of the radii of curvature. Thus, Channel 1 is termed a sharp bend, and Channel 2 is termed a shallow bend.

The experimental study of Edwards et al. [42] investigated shock propagation in the channels and reported the speed of the shock waves, as well as the corresponding Mach numbers at the inner and outer walls. Figure 5.2 is a sketch of the main features of a shock wave traveling in the channels. S,R,T,H, C and M_0 label these main features. Once the initially planar shock wave forms in the straight section of the tube, it propagates into the curved section of the channel. This results in the generation of flow waves, as sketched in Figure 5.2. These waves meet at the point, T, which is termed the triple or shock-shock point [44]. Two incident waves are to the right of the triple point while to the left is a reflected wave (R) and a contact discontinuity (S). The contact discontinuity separates the gas passing through the Mach stem (TC) and the incident shock (ET) [44]. Mach reflection and its associated Mach stem arise in the reflection

of strong shock [10]. The reflected shock is the curve (RT). The contact discontinuity (ST) is placed behind the original incident shock wave (ET). This shock wave weakens along inner wall, while the Mach stem strengthens along the outer wall.



(a) Early propagation of the incident shock wave

(b) Shock propagation downstream of the channel

Figure 5.2: Sketch of the traveling shock wave in air towards the end of the sharp tube (Channel 1) with a 90° bend. This figure is taken from [42].

Figure 5.2a displays the early propagation of the shock wave. There is still a remnant of the initial shock wave, which is denoted by M_0. On the further propagation, Figure 5.2b, the initial shock wave disappears. This is due to expansion waves, which are created along the inner wall, reaching the triple point. This is the reason for the reduction in strength of the shock wave from its initial value, M_0, and its attenuation along the inner wall. However, while the incident shock wave loses its strength along the inner wall, it is getting stronger along the outer wall.

There is a slight variation in the development of the shock curvature in the downstream field between the sharp and shallow bends. Although the curvature of the shock front retains the form in Channel 1 as illustrated in Figure (5.2), it is not the same case for Channel 2. To this end, the propagation of a shock for the initial Mach number $M_0 = 2.9$ is illustrated in the midportion of Channel 2 in Figure 5.3. Since the triple point in Channel 2 moves towards the inner wall faster than in Channel 1, the interaction of the triple point with the expansion waves, which are created at the inner walls, occurs before or right after the end of the curved section of Channel 2 depending on the magnitude of the initial Mach number. Therefore, the shock wave regains the planar form as at the beginning of Channel 2. All of these physical features, which are

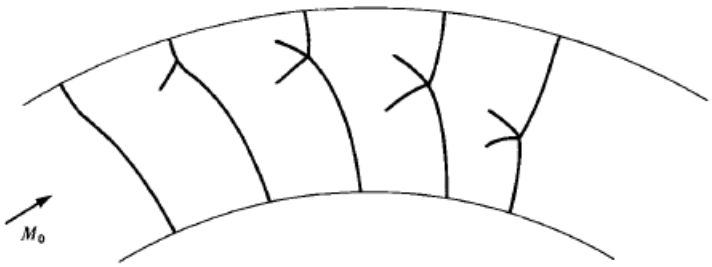


Figure 5.3: Sketch of the traveling shock wave in air along the midportion of the shallow tube (Channel 2) with a 90° bend at $M_0 = 2.9$. This figure is taken from [42].

created in the flow field and around the shock waves, are obtained for both channels by numerically solving the nonlinear hyperbolic system of the Euler equations of gas dynamics in polar coordinates. A comparison of the experimental results of Edwards et. al. [42] and numerical solutions obtained in this chapter is presented at the end of this chapter.

5.2 The Euler equations of gas dynamics in polar coordinates

The nonlinear two-dimensional Euler equations of gas dynamics in polar coordinates are [90] ,

$$\begin{aligned}
\rho_t + \tfrac{1}{r}(r\rho u_r)_r + \tfrac{1}{r}(\rho u_\theta)_\theta &= 0 \\
(\rho u_r)_t + \tfrac{1}{r}(r\rho u_r^2)_r + \tfrac{1}{r}(\rho u_r u_\theta)_\theta &= \tfrac{\rho u_\theta^2}{r} - p_r \\
(\rho u_\theta)_t + \tfrac{1}{r}(r\rho u_r u_\theta)_r + \tfrac{1}{r}(\rho u_\theta^2)_\theta &= -\tfrac{\rho u_\theta u_r}{r} - \tfrac{1}{r}p_\theta \\
E_t + \tfrac{1}{r}(r u_r(E+p))_r + \tfrac{1}{r}(u_\theta(E+p))_\theta &= 0,
\end{aligned} \tag{5.1}$$

where ρ is the density, u_r and u_θ are the velocities in the r and θ directions, p is the pressure, and E is the total energy. The subscripts, except those for velocities, refer to first and second derivatives with respect to time, t, radius, r and angle, θ. Numerical methods for solving the compressible gas equations normally generate nonphysical oscillations behind a shock [61]. To eliminate these equation (5.1) needs to be modified by adding artificial viscosity to the gas equations. Therefore, the following scalar and

vectoral Laplace operators are introduced for the Euler fluxes in polar coordinates [90]

$$\begin{aligned}
\nabla^2\rho &= \left(\tfrac{1}{r}\rho_r + \rho_{rr} + \tfrac{1}{r^2}\rho_{\theta\theta}\right) \\
\nabla^2\rho u_r &= \left((\rho u_r)_{rr} + \tfrac{1}{r}(\rho u_r)_r + \tfrac{1}{r^2}(\rho u_r)_{\theta\theta} - \tfrac{\rho u_r}{r^2} - \tfrac{2}{r^2}(\rho u_\theta)_\theta\right) \\
\nabla^2\rho u_\theta &= \left((\rho u_\theta)_{rr} + \tfrac{1}{r}(\rho u_\theta)_r + \tfrac{1}{r^2}(\rho u_\theta)_{\theta\theta} - \tfrac{\rho u_\theta}{r^2} + \tfrac{2}{r^2}(\rho u_r)_\theta\right) \\
\nabla^2 E &= \left(\tfrac{1}{r}E_r + E_{rr} + \tfrac{1}{r^2}E_{\theta\theta}\right),
\end{aligned} \tag{5.2}$$

where $\nabla^2\rho$ and $\nabla^2 E$ are scalar Laplace operators, and $\nabla^2\rho u_r$ and $\nabla^2\rho u_\theta$ are vectoral Laplace operators, respectively. The artificial viscosities for the gas equations are obtained by multiplying the Laplace operators by artificial diffusion constants, $AD_\rho = \epsilon_1\nabla^2\rho$, $AD_{\rho u_r} = \epsilon_2\nabla^2\rho u_r$, $AD_{\rho u_\theta} = \epsilon_3\nabla^2\rho u_\theta$ and $AD_E = \epsilon_4\nabla^2 E$. Here, $\epsilon_i,\ i = 1,\ldots,4$ are small enough constants to suppress non-physical overshoots or undershoots resulting in nonphysical oscillations, which can potentially be caused by numerical solutions.

The gas equations with artificial viscosity are

$$\begin{bmatrix}\rho\\ \rho u_r\\ \rho u_\theta\\ E\end{bmatrix}_t + \frac{1}{r}\begin{bmatrix}r\rho u_r\\ r(\rho u_r^2 + p)\\ r\rho u_r u_\theta\\ r u_r(E+p)\end{bmatrix}_r + \frac{1}{r}\begin{bmatrix}\rho u_\theta\\ \rho u_r u_\theta\\ \rho u_\theta^2 + p\\ u_\theta(E+p)\end{bmatrix}_\theta = \frac{1}{r}\begin{bmatrix}0\\ \rho u_\theta^2 + p\\ -\rho u_r u_\theta\\ 0\end{bmatrix} + \begin{bmatrix}\mathrm{AD}_\rho\\ \mathrm{AD}_{\rho u_r}\\ \mathrm{AD}_{\rho u_\theta}\\ \mathrm{AD}_E\end{bmatrix}. \tag{5.3}$$

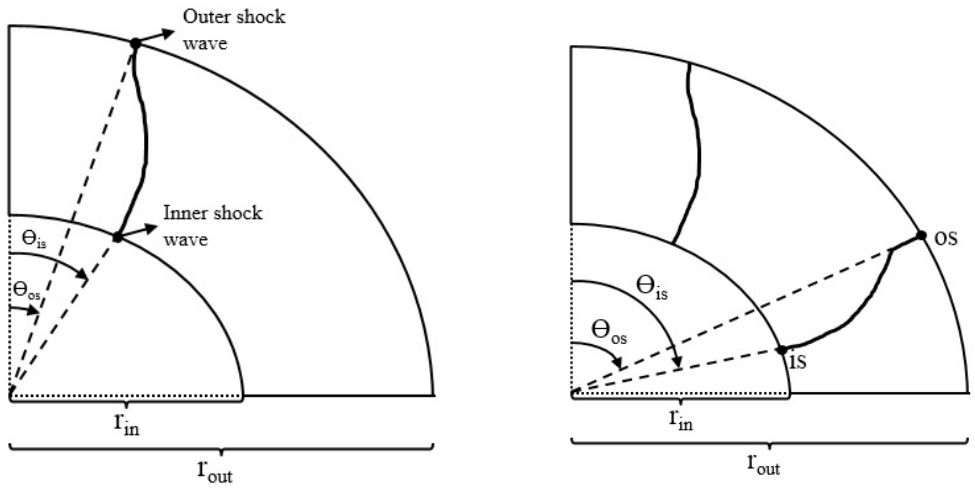
A similar form to equations (5.3) was used in [91] with the artificial viscosity terms neglected.

The Mach numbers at the inner and outer walls are calculated by dividing the velocity of the shock by the speed of sound ahead of the shock so that

$$\begin{aligned}
M_{wi} &= \tfrac{V_{si}}{a_R} \\
M_{wo} &= \tfrac{V_{so}}{a_R}.
\end{aligned} \tag{5.4}$$

Here, $V_{si},\ M_{wi}$ and $V_{so},\ M_{wo}$ are the velocities and wall Mach numbers of the shock

at the inner and outer walls, respectively. The term $a_R = \sqrt{\frac{\gamma p_R}{\rho_R}}$ is the speed of sound ahead of the shock wave. p_R is the initial pressure and ρ_R is the initial density. These quantities are in the region where the shock wave has not reached. Since the speed of sound is known, there is a need to calculate the velocity of the shock wave at both walls. First, the position of shock on inner and outer walls has to be found. This position indicates the current position of the shock wave, as illustrated in Figure (5.4), at a given time. The angle for this position is stored as a vector to be used in the calculation of the velocity of shock wave. The strategy developed for the determination of this



(a) The position of inner and outer shock wave

(b) Inner and outer shock propagation downstream of the channel

Figure 5.4: Sketch of the traveling distance of the shock waves along the inner and outer walls. Here, θ_{is} and θ_{os} are position angles of inner and outer wall shock waves.

position relies on the calculation of the maximum absolute value of the derivative of density with respect to the angle at the wall θ. This procedure is repeated along the inner and outer walls at every time step. The Matlab code for this procedure is given in Appendix C. After finding the angles of shock wave at the inner and outer walls, the calculation continues by finding the values of velocity of the shock wave at the inner and outer walls.

The velocity of any wave is calculated from the ratio of total distance traveled over the total time spent. Therefore, the velocity of the shock wave is computed by

$$Vs_i = \frac{r_i d\theta_{\mathrm{is}}}{dt} \tag{5.5}$$

and

$$Vs_o = \frac{r_o d\theta_{\text{os}}}{dt}, \qquad r_o > r_i, \tag{5.6}$$

where θ_{is} and θ_{os} are the position angles of the shock waves at the inner and outer walls, respectively. Here, $r_i d\theta_{\text{is}}$ and $r_o d\theta_{\text{os}}$ are the total distances traveled along the inner and outer walls, respectively, in time dt. Vs_i and Vs_o are the velocities of the shock at the inner and outer walls, respectively. Since the numerical results obtained in this work are compared with the experimental data reported in Edwards et al. [42], there should be consistency between the units used for these results. In the experimental work [42], $M_{\{wi,wo\}}$ numbers were normalized using the incident shock number, M_0 because the physical shock tube was a combination of two straight tubes and a channel with a 90° bend for two different dimensions. Therefore, Mach numbers $M_{\{wi,wo\}}$ are also scaled by M_0 in this study. Thus, the Mach numbers equation (5.4) are divided by the incident Mach number, M_0 to become

$$\begin{aligned} M_{Ni} &= \tfrac{M_{wi}}{M_0} \\ M_{No} &= \tfrac{M_{wo}}{M_0}, \end{aligned} \tag{5.7}$$

where M_{Ni} is the normalized inner wall Mach number and M_{No} is the normalized outer wall Mach number. M_0 varies between $1.2 \leq M_0 \leq 2.9$ for Channel 1 and Channel 2 [42]. Furthermore, the lengths of the inner and outer walls were also non-dimensionalized in the experimental study [42]. Following this non-dimensionalization, we set

$$\begin{aligned} X_{Ni} &= \tfrac{r_{in}\theta^i_{in}}{r_{out}-r_{in}} \\ X_{No} &= \tfrac{r_{out}\theta^i_{out}}{r_{out}-r_{in}}, \end{aligned} \tag{5.8}$$

where X_{Ni} and X_{No} are the non-dimensionalized inner and outer wall lengths, r_{in} and r_{out} are the radii of inner and outer walls. θ^i_{in} and θ^i_{out} are the angles of the shock at the inner and outer walls.

5.2.1 Spatial Discretization for 2D : CCFVM

In Chapter 4, the cell-centered finite volume (CCFVM) was studied in detail for the one dimensional compressible inviscid hyperbolic Euler equations. Here, the CCFVM

is applied to discretize the two dimensional Euler equations of gas dynamics in polar coordinates equations (5.1). The computational domain was structured by computational cells in the one dimensional case and is built on sub-finite computational grid cells in the two dimensional case. These are defined by

$$D_{i,j} = [\theta_{i-1/2,j}, \theta_{i+1/2,j}] \times [r_{i,j-1/2}, r_{i,j+1/2}], \tag{5.9}$$

where $\theta_{i\pm1/2,j} = (\theta_{i,j} + \theta_{i\pm1,j})/2$ and $r_{i,j\pm1/2} = (r_{i,j} + r_{i,j\pm1})/2$ are the mid points of the cell edges. Since the finite volume method is based on an approximation to the integral form of the conservation laws, the numerical solution variables for the two dimensional problem are obtained by using the following discretized equation for each (i,j)th cell

$$u_{i,j} \approx \frac{1}{\Delta\theta}\frac{1}{\Delta r}\int\int_{D_{i,j}} u(\theta, r, t)\, d\theta dr. \tag{5.10}$$

Here, $\Delta r = r_{i,j+1/2} - r_{i,j-1/2}$ and $\Delta\theta = \theta_{i+1/2,j} - \theta_{i-1/2,j}$ are the lengths of spatial meshes in the radial, r, and angular, θ, spatial directions, respectively. At this stage, the two dimensional Euler equations in polar coordinates can be stated in integral form over each computational grid cell $D_{i,j}$ by

$$\begin{aligned} \frac{\partial}{\partial t}\int\int_{D_{i,j}} u(\theta, r, t)\, d\theta dr \quad &= \int_{r_{i-1/2}}^{r_{i+1/2}} \left(f(u(\theta_{i-1/2}, r, t)) - f(u(\theta_{i+1/2}, r, t))\right) dr \\ &+ \int_{\theta_{i-1/2}}^{\theta_{i+1/2}} \left(g(u(\theta, r_{i-1/2}, t)) - f(u(\theta, r_{i+1/2}, t))\right) d\theta, \end{aligned} \tag{5.11}$$

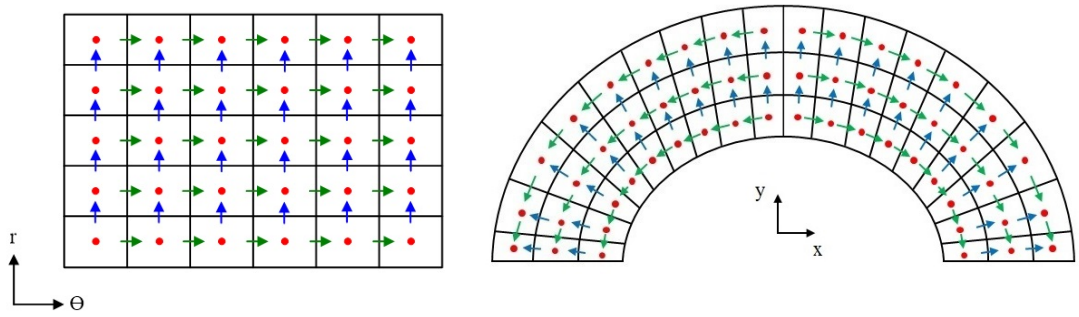
where the solution vector u and corresponding flux vectors f and g are stated in equations (5.1). The numerical coding process for the gas equation is done in two steps to prevent any possible boundary issues created when the CCFVM method is applied. First, the homogeneous Euler equations neglecting the source and artificial diffusion terms are discretized. Next, diffusion and source terms are included. The integrated form of the homogeneous governing Euler equations is interpreted by the following discretization procedure as

$$\frac{d}{dt}u_{i,j}(t) = -\frac{1}{\Delta\theta}\left[f_{i+1/2,j} - f_{i-1/2,j}\right] - \frac{1}{\Delta r}\left[g_{i,j+1/2} - g_{i,j-1/2}\right]. \tag{5.12}$$

The following numerical flux functions are used to approximate the integral form of the fluxes as

$$\begin{aligned} f_{i\pm1/2,j} = \frac{f_{i\pm1,j}+f_{i,j}}{2} &\approx \frac{1}{\Delta r}\int_{r_{i-1/2}}^{r_{i+1/2}} f(u(\theta_{i\pm1/2}, r, t))dr \\ g_{i,j\pm1/2} = \frac{g_{i,j\pm1}+g_{i,j}}{2} &\approx \frac{1}{\Delta\theta}\int_{\theta_{i-1/2}}^{\theta_{i+1/2}} g(u(\theta, r_{i\pm1/2}, t))d\theta, \end{aligned} \tag{5.13}$$

where the f and g fluxes are in the θ and r directions, respectively. A sketch of the computational and physical domains for interior parts is given in Figure 5.5. The f



(a) Computational Domain **(b)** Physical Domain

Figure 5.5: The sketch of the domains for interior scheme.

fluxes (colored with green arrows) move in the θ direction, the g fluxes (colored with blue arrows) move in the r direction. In addition, the red points represent the cell average discretized solution variables for each (i, j) cell. The index values for each computational grid cell varies between $i = 1, \ldots, M$ and $j = 1, \ldots, N$. Here, the main focus is given to the interior part of the numerical scheme, whereas the boundaries, for which $i = 1, M$ and $j = 1, N$, are left to be discussed in section 5.2.4. The initial discussion of the CCFVM method for the one dimensional case was presented in Chapter 4. The necessary artificial viscosities are now applied to the two dimensional discretized fluxes for equations (5.3). These fluxes are placed in the numerical scheme as

$$\frac{d}{dt}u_{i,j}(t) = -\frac{1}{\Delta\theta}\left[f^{\star}_{i+1/2,j} - f^{\star}_{i-1/2,j}\right] - \frac{1}{\Delta r}\left[g^{\star}_{i,j+1/2} - g^{\star}_{i,j-1/2}\right], \tag{5.14}$$

where

$$f^{\star}_{i+1/2,j} = \frac{f_{i+1,j} + f_{i,j}}{2} - \epsilon_1 \frac{u_{i+1,j} - u_{i,j}}{\Delta\theta} \tag{5.15}$$

and

$$g^{\star}_{i,j+1/2} = \frac{g_{i,j+1} + g_{i,j}}{2} - \epsilon_2 \frac{u_{i,j+1} - u_{i,j}}{\Delta r}. \tag{5.16}$$

Here $f^{\star}_{i\pm1/2,j}$ and $g^{\star}_{i,j\pm1/2}$ are modified fluxes and ϵ_1 and ϵ_2 are the small positive artificial diffusion coefficients. However, equivalent form of these modified fluxes are required to simplify the application of the boundary conditions to the numerical scheme. Thus, the final forms of the modified numerical fluxes are

$$f^{\star}_{i\pm1/2,j} = \frac{f_{i\pm1,j} + f_{i,j}}{2} - \epsilon_1 \left(\frac{\partial u}{\partial \theta}\right)_{i\pm1/2,j} \tag{5.17}$$

and

$$g^{\star}_{i,j\pm1/2} = \frac{g_{i,j\pm1} + g_{i,j}}{2} - \epsilon_2 \left(\frac{\partial u}{\partial r}\right)_{i,j\pm1/2}. \tag{5.18}$$

After implementing equations (5.17) and (5.18) into the modified equation (5.14), the homogeneous fluxes with artificial viscosity terms are

$$\frac{d}{dt}u_{i,j}(t) = -\frac{1}{\Delta\theta}\left[f_{i+1/2,j} - f_{i-1/2,j}\right] + \epsilon_1 \left(\frac{\partial^2 u}{\partial \theta^2}\right)_{i,j} - \frac{1}{\Delta r}\left[g_{i,j+1/2} - g_{i,j-1/2}\right] + \epsilon_2 \left(\frac{\partial^2 u}{\partial r^2}\right)_{i,j}. \tag{5.19}$$

Here, the artificial diffusion terms are given by

$$\left(\frac{\partial^2 u}{\partial \theta^2}\right)_{i,j} = \frac{\left(\frac{\partial u}{\partial \theta}\right)_{i+1/2,j} - \left(\frac{\partial u}{\partial \theta}\right)_{i-1/2,j}}{\Delta\theta} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta\theta^2} \tag{5.20}$$

and

$$\left(\frac{\partial^2 u}{\partial r^2}\right)_{i,j} = \frac{\left(\frac{\partial u}{\partial r}\right)_{i,j+1/2} - \left(\frac{\partial u}{\partial r}\right)_{i,j-1/2}}{\Delta r} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta r^2}. \tag{5.21}$$

Finally, the governing Euler equations with artificial diffusion and source terms are

$$\begin{aligned}\frac{d}{dt}u_{i,j}(t) = &-\frac{1}{\Delta\theta}\left[f_{i+1/2,j} - f_{i-1/2,j}\right] + \epsilon_1 \left(\frac{\partial^2 u}{\partial \theta^2}\right)_{i,j} - \frac{1}{\Delta r}\left[g_{i,j+1/2} - g_{i,j-1/2}\right] + \epsilon_2 \left(\frac{\partial^2 u}{\partial r^2}\right)_{i,j} \\ &+ GS_{i,j}.\end{aligned} \tag{5.22}$$

Here, $GS_{i,j}$ represents the discretized source terms for the momentum equation in the r direction, $\frac{\rho_{i,j} u_{\theta^2_{i,j}}}{r_{i,j}} - \left(\frac{\partial p}{\partial r}\right)_{i,j}$, and for the momentum equation in the θ direction,

$-\frac{\rho_{i,j}u_{\theta i,j}u_{r\,i,j}}{r_{i,j}} - \frac{1}{r_{i,j}}\left(\frac{\partial p}{\partial \theta}\right)_{i,j}$, as in equations (5.1). The derivatives of the pressure in terms of radial and angular variables are derived in the same way using the flux functions

$$\left(\frac{\partial p}{\partial r}\right)_{i,j} = \frac{p_{i,j+1/2} - p_{i,j-1/2}}{\Delta r} \tag{5.23}$$

and

$$\left(\frac{\partial p}{\partial \theta}\right)_{i,j} = \frac{p_{i+1/2,j} - p_{i-1/2,j}}{\Delta \theta}. \tag{5.24}$$

Since the governing equations are unsteady the updated numerical solutions at each time step are obtained by applying the time discretization method.

5.2.2 Temporal Integration for 2D:SSPRK

The flow develops in time as well as in the radial and angular spatial variables. The implementation of the time discretization method is developed as the third order strong stability preserving Runge-Kutta (SSPRK) method for the two dimensional Euler equations in polar coordinates, equations (5.1), as detailed in the one dimensional case in Chapter 4. The major advantage of the use of a strong stability preserving Runge-Kutta scheme is not only the prevention of potential stability issues, but also its convenient application to multidimensional PDE problems [86]. The only difference for the implementation of the SSPRK method to the two-dimensional case compared with the one-dimensional case is the value of the CFL number. This is defined as

$$CFL = \frac{dt}{\min\left(\frac{rd\theta}{\max|u_\theta + a|}, \frac{dr}{\max|u_r + a|}\right)}, \tag{5.25}$$

where $rd\theta$ and dr are length of the distance traveled in time dt in the θ and r directions, respectively. Also, $\max|u_\theta + a|$ and $\max|u_r + a|$ are the maximum eigenvalues of fluxes in the θ and r directions achieved by their Jacobians, i.e., $f_u(u)$ and $g_u(u)$, respectively. The stability criterion requires that we halve the CFL number for the one-dimensional case as stated in [86]. Hence, it is taken to be 0.5 for the numerical solutions of this chapter.

The two dimensional Euler equations of gas dynamics in polar coordinates are augmented by appropriate initial and boundary conditions. These conditions are explained

in Sections 5.2.3 and 5.2.4, respectively.

5.2.3 Initial Conditions

The initial state is the fluid at rest at the beginning of curved part of the channel. The gas in the left section of the computational domain is at a higher pressure and has a greater density than those in the right section of computational domain. The experiments of Edwards at al. [42] reported the initial Mach number of the shock. The initial pressure and density in the left of the domain were then determined from the shock jump conditions based on this Mach number. The shock conditions in terms of this upstream Mach number given by Whitham [10] are then used to find the gas state behind the initial shock. These give

$$M = \frac{U - u_1}{a_1} = \left(1 + \frac{\gamma + 1}{2\gamma} z\right)^{1/2}, \tag{5.26}$$

$$\frac{\rho_2}{\rho_1} = \frac{1 + \frac{\gamma+1}{2\gamma} z}{1 + \frac{\gamma-1}{2\gamma} z}, \tag{5.27}$$

where $z = (p_2 - p_1)/p_1$ is the shock strength. The known values of velocity u_1, density ρ_1 and sound speed a_1 are for the region of gas ahead of the shock wave. M is the initial Mach number and U is velocity of the shock wave. The remaining variables are the density, ρ_2, and pressure, p_2, of the gas, in the region of the gas behind of the shock wave. However, the values of density and pressure ahead of the shock are not given in the work of Edwards et al. [42]. Here, three different initial states are examined for each channel, as in Edwards et al. [42]. Mach numbers 1.7, 2.1 and 2.7 are used for Channel 1 and Mach numbers 1.2, 1.9 and 2.9 are considered for Channel 2, as in the experimental work [42]. Given these initial Mach numbers, the unknown state in the region behind the shock wave can be deduced once the state ahead is known by using equations (5.26) and (5.27). The state ahead of the shock was assumed to be at standard temperature and pressure. The initial values for the region ahead of the shock waves was then computed by using the ideal gas law, $p = \rho RT$ [10]. Here, the values R and T are the gas constant, 287.058 $Jkg^{-1}K^{-1}$, and the temperature is $20°C = 293\ K$.

The pressure is $p = 1\ atm = 101325\ Pa$. We then have

$$p_1 = 101325\ \ Pascal\ \ (1\ atm) \tag{5.28}$$

$$\rho_1 = 1.2043\ \ kg/m^3 \tag{5.29}$$

$$u_1 = 0\ \ m/s \tag{5.30}$$

$$v_1 = 0\ \ m/s. \tag{5.31}$$

By using these initial values for the region ahead of the initial shock, the initial state behind the shock can be determined via equations (5.26) and (5.27) for each Mach number for both channels.

Channel 1:

Case 1

The initial Mach number is $M = 1.7$ for Case 1. The shock strength z is calculated using equation (5.26). z is calculated to be 2.2050 for $M = 1.7$. We then have

$$z = (p_2 - p_1)/p_1$$

$$p_2 = (1 + z)p_1$$

$$p_2 = 3.2050p_1.$$

So that $p_2 = 324750\ Pa$ and $\rho_2 = 2.6467\ kg/m^3$ on using the jump condition (5.27). Since the gas is rest at the initial state the velocities are equal to zero, $u_2 = 0$ and $v_2 = 0$.

Case 2

As in Case 1, the following initial values are calculated for the behind the initial shock region when the Mach number equals to 2.1

$$z = 3.9783$$

$$p_2 = 4.9783p_1$$

$$\rho_2 = 2.8119\rho_1.$$

From equations (5.28) and (5.29) we found p_2 and ρ_2 to be 504430 Pa and 3.3864 kg/m^3, respectively.

Case 3

The final case has the initial Mach number of 2.7. The relation between the pressures and densities is determined by following the same procedure stated for Cases 1 and 2, so that

$$z = 7.3383$$

$$p_2 = 8.3383 p_1$$

$$\rho_2 = 3.5590 \rho_1.$$

As a result, p_2 is 844880 Pa, and ρ_2 is 4.2861 kg/m^3.

Channel 2:

Case 1

The first case for Channel 2 has the initial Mach number 1.2. The state behind the shock is calculated from that ahead. The same procedure is used to calculate the initial state for Channel 2 as for Channel 1. The initial values to the left of the shock are

$$z = 0.5133$$

$$p_2 = 1.5133 p_1$$

$$\rho_2 = 1.3416 \rho_1.$$

Using the values given in equations (5.28) and (5.29) for the above equations, p_2 and ρ_2 are 153349 Pa and 1.6157 kg/m^3, respectively.

Case 2

In case 2, the initial Mach number is 1.9 for the following numerical simulations of Channel 2. We then have

$$z = 3.0450$$

$$p_2 = 4.0450 p_1$$

$$\rho_2 = 2.5157 \rho_1.$$

So that $p_2 = 409860\ Pa$ and $\rho_2 = 3.0297\ kg/m^3$.

Case 3

The greatest initial Mach for Channel 2 is $M = 2.9$. Hence

$$z = 8.6450$$

$$p_2 = 9.6450 p_1$$

$$\rho_2 = 3.7629 \rho_1.$$

Then p_2 and ρ_2 are calculated to be 977280 Pa and 4.5317 kg/m^3, respectively.

5.2.4 Boundary Conditions and Ghost Cells

Boundary conditions are essential to fully determine the flow. In our numerical simulation two types of boundary conditions are used around the computational domain, as illustrated in Figure 5.6. It is worth noting that the left curved region is just used for convenience and does not exist in the experiments. Therefore, the numerical solutions are demonstrated solely for the right curved region. It needs to be highlighted that since all initial Mach numbers are greater than 1, the gas flow is supersonic. The direction of gas fluid is towards the downstream of the bend and this type of flow should be supported by the numerical and physical boundary conditions. Outflow boundary conditions are used to be implemented at two ends of the computational domain [92]. This means that solution values from interior can be extrapolated to the exits of the domain. The essential point here is the signs of the characteristics, which are the eigenvalues of the Jacobian fluxes, i.e., $\partial f/\partial u_\theta$ and $\partial g/\partial u_r$, of the governing system.

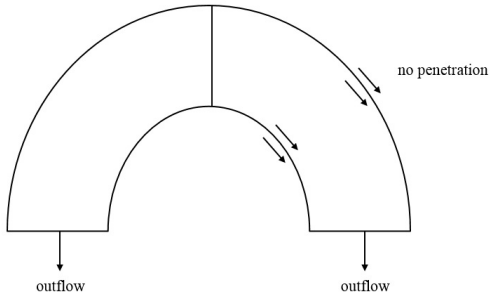


Figure 5.6: Sketch of numerically implemented boundary conditions at the beginning, at the end, as well as the side walls of the channel.

While the number of physical boundary conditions is given by the number of positive characteristics, the number of the numerical boundary conditions is provided by the negative characteristics.

The numerical boundary conditions for the numerical simulation have been implemented by enlarging the computational domain with an appropriate number of additional cells, which are so-called ghost cells [50]. These cells contain values which depend on the values from interior points and appropriate boundary conditions. In our numerical calculations, the boundary conditions are set as *outflow boundary conditions* at the relevant end of the both channels. The main reason to choose that is to prevent any possible generation of spurious boundary problems into the domain at the end of the tube and to allow outgoing gas to flow without any disturbance at the exit [18]. To assign values to the ghost cells according to the outflow boundary condition at $\theta = \pi/2$ for $r_{in} \leq r \leq r_{out}$, the interior cell values are extrapolated into the ghost cells. Since the flow is supersonic, there are four negative eigenvalues of f and g fluxes. These eigenvalues are such that

$$\lambda_1^f = u_\theta, \quad \lambda_2^f = u_\theta, \quad \lambda_3^f = u_\theta + a, \quad \lambda_4^f = u_\theta - a,$$

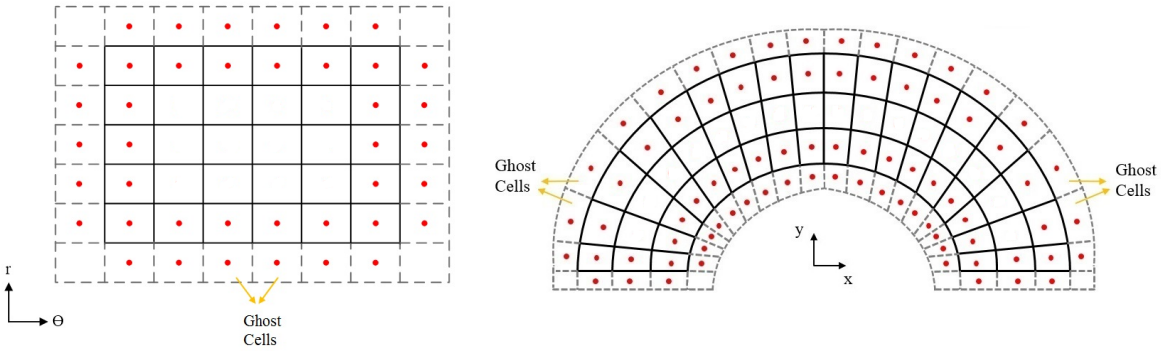
$$\lambda_1^g = u_r, \quad \lambda_2^g = u_r, \quad \lambda_3^g = u_r + a, \quad \lambda_4^g = u_r - a,$$

where λ^f and λ^g are the eigenvalues of Jacobian fluxes of f and g, respectively, and a is the speed of sound. The extrapolated number of values are determined by the number of these negative eigenvalues of the system. Therefore, the ghost cell average values are

obtained from the following four flow variables

$$\rho_{M+1,j} = \rho_{M,j}, \quad p_{M+1,j} = p_{M,j}, \quad u_{rM+1,j} = u_{rM,j}, \quad u_{\theta M+1,j} = u_{\theta M,j}, \tag{5.32}$$

where $j = 1 \dots N$. These values are used for a single ghost cell as shown for both the computational and physical domains in Figure 5.7. It should be noted that the



(a) Computational domain with one ghost cell

(b) Reflection of the ghost cell procedure onto the physical domain

Figure 5.7: The sketch of the extended domains with one ghost cells (grey dashed lines) around of entire domains.

ghost cells are not placed at the entrance of the channel (see Figure 5.7). The reason is that only numerical boundary conditions require the ghost cells, and only physical boundary conditions are needed at the inlet of the supersonic flow region. If a second set of ghost cells is needed, the same procedure is repeated. Since only one ghost cell is employed in the numerical method in our work, the second cell average ghost cell is not used in the computations. After explaining how to find flow values in the ghost cells for the outflow boundary conditions, the boundary conditions for the side walls will be described.

In the experiments the sides of the channels were impermeable. Therefore, the boundary conditions are considered as *no penetration* or *solid wall boundary conditions* so that the normal velocity in the r direction is zero, $\mathbf{u}_r.\mathbf{n} = 0$ [18]. There are two walls to be taken into consideration when setting the flow values in ghost cells. They are at the inner and outer walls at $r = r_{in}$ and $r = r_{out}$, respectively. While these values are measured as $r_{in} = 48.9\ mm$ and $r_{out} = 101.1\ mm$ for Channel 1, they are $r_{in} = 123.9\ mm$ and $r_{out} = 176.1\ mm$ for Channel 2. The cell average values for ghost

cells are only given for one wall here as the same procedure is applied at the other wall. In ghost cells, the flow quantities are set according to equations (5.32), except the velocities in the r direction. The velocities in the ghost cells at the inner and outer walls are implemented in the numerical method by negating the cell average interior solution values

$$u_{ri,0} = -u_{ri,1}, \quad u_{ri,N+1} = -u_{ri,N}, \quad i = 1, \ldots, M. \tag{5.33}$$

The main reason for changing the velocity component in the r direction is that only one positive eigenvalue exists for solid wall boundary conditions and one physical boundary condition needs to be applied to the numerical equations [92]. By setting these values as in equation (5.33), the shock waves can bounce from the boundaries and travel towards the opposite wall.

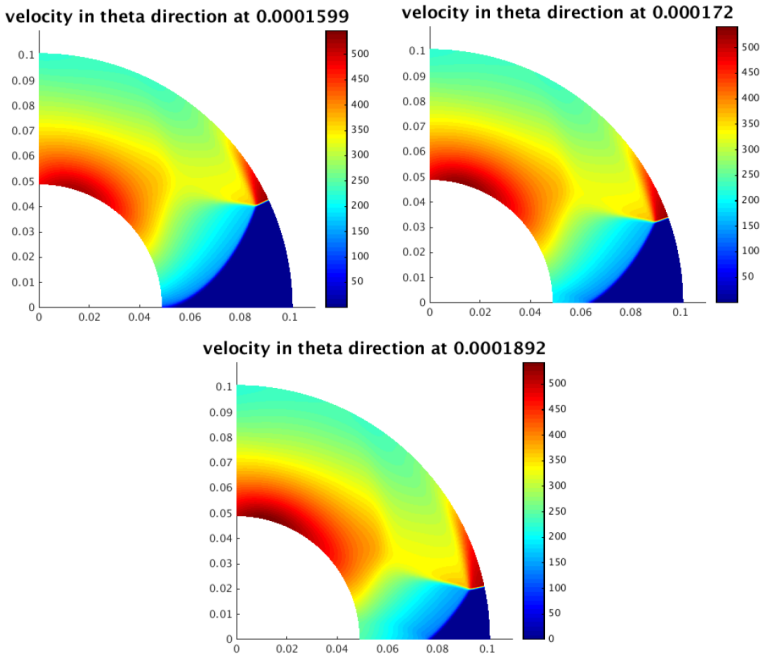


Figure 5.8: Illustration of the effect of outflow and no-penatration wall boundary conditions for the sharp bend for the initial Mach number $M_0 = 2.7$.

The robustness of the proposed boundary treatment is illustrated with one of the cases with initial Mach number 2.7 for Channel 1. The numerical solution is shown in Figure 5.8. Here, the propagation of the shock is in agreement with the experimental

profile shown in Figure 5.2. The incident shock Mach stem and reflected shock can be clearly seen as can the expansion waves. It can be seen that there is no unphysical inflow or reflected at the exit. The result of this numerical simulation proves that the two boundary conditions, i.e., outflow and no-penetration, are accurately implemented into the discretized governing equations. The outflow boundary conditions allow any outgoing flows to pass through the boundaries smoothly. Moreover, the flow is not allowed to pass through the side walls, and reflections from inner and outer walls are seen as yellow waves in Figure 5.8. Consequently, the numerical computational process is constructed accurately and perfect agreement is achieved with the experimental study [42].

5.3 Numerical Simulations and Results

The experimental study of shock wave diffraction in two different channels with 90° bends was performed by Edwards et al. [42]. In their study, the first channel (Channel 1) had a sharp bend, and its inner and outer radii were 48.9 mm and 101.1 mm, respectively (see Figure 5.1). For this channel three different initial Mach numbers were set for initial planar shock wave at the beginning of the curvature of Channel 1. These Mach numbers ranged from 1.7 to 2.7. The second channel (Channel 2) had a shallow bend and its inner and outer radii were 123.9 mm and 176.1 mm, respectively. Three different Mach numbers ranging from 1.2 to 2.9 were used to examine how the initial planar shock waves change while moving along the channels.

In the present work, the propagation of the shock waves around the bend and the wall Mach numbers for both the inner and outer walls were calculated numerically. Then, the numerical solutions were compared with the experimental results to show the robustness of the numerical scheme. It is worth to note that in [43] numerical solutions for the shock wave propagation were studied using geometrical shock dynamics based on Whitham's theory [10]. However, the flow field behind the shock wave is not determined by this theory of geometrical shock dynamics as it only considers the propagation of the shock treating it as a front. In this work, numerical solutions for the shock motion and the flow filed behind the shock wave are obtained by solving the nonlinear gas dynamics Euler equations in polar coordinates, equations (5.1), numerically. In the numerical

calculations, the CCFVM and the SSPRK3 numerical methods were used for the space and time discretizations, respectively. The governing system of PDEs was augmented with the appropriate initial and boundary conditions. The initial conditions for flow parameters, density, pressure and velocities, are determined for each Mach number and for each channel as explained in Section 5.2.3. Appropriate boundary conditions were implemented into the numerical scheme based on the numerical and physical features of the flow, as well as the shape of the channels. These outflow and no penetration wall boundary conditions were also explained explicitly in Section 5.2.4.

Now, numerical solutions are given for Channel 1 with all three different initial Mach numbers, M_0, 1.7, 2.1, 2.7, as in the experimental study [42]. In addition, all the data in the reported experimental plots in [42] for the inner and outer Mach wall numbers were converted into numbers by using Engauge software [93]. Since the results were published in the earlier years, the represented results were primarily screenshotted from the original study [42] and then saved as JPEG file. This JPEG file was then imported into the Engauge digitizer software to transfer the data available on the image into a Microsoft Excel file. The outcomes are the desired Mach numbers of shock waves on the outer and inner walls and are used in the Matlab code during the comparison between numerical and experimental data.

Case 1 : $\mathbf{M_0 = 1.7}$

The numerical computations were conducted for a 90° sharp bend for an initial Mach number of 1.7. The computational domain in the r and θ directions was divided into 100x400, 200x800 and 400x1600 uniform structured computational domains with the final time, $T_{end} = 0.0001944\ s$, as illustrated in Figure 5.5a. An initial planar shock wave is placed at the beginning of the curvature of the channel at $\theta = 0°$ and $r_{in} \leq r \leq r_{out}$. The CFL constant is taken as 0.5 and the artificial diffusion coefficients were chosen small enough to dampen any possible non-physical numerical oscillations around the shock waves. The diffusion coefficients vary according to the size of the computational cells and were 0.1, 0.04 and 0.03 for all three different size domains, 100x400, 200x800 and 400x1600, respectively.

In Figure 5.9, the solutions for the flow and the propagation of the shock front are displayed as a contour plot for the velocity in the θ direction. The numerical

solution is illustrated for every 10° angles for 400x1600 computational domain. The evolution of the initial planar shock wave towards downstream of Channel 1 is clear. The propagation of the shock front is clear as the development of the reflected waves in the flow field behind the shock front. In Figure 5.9, the numerical simulations lead to visualize the all the important formed physical features behind of the shock waves while it is moving downstream of channel, i.e., position of the triple point, reflected yellow waves from the inner and outer walls, the Mach stem, the attenuated curved shock wave and the contact discontinuity. However, these features can be visualized more clearly for a stronger initial Mach, $M_0 = 2.7$, as shown in Figure 5.16. It is worth to note that experiments merely provide the information about the shock front but not about the flow field behind of it.

In Figures 5.10 and 5.11, numerical evolution results are depicted for the density and pressure flow variables. The evolution of these quantities can be interpreted by using the ideal gas law $p = \rho RT$. The effect of any change in the flow quantities is observed from the amount of compressed gas at the outer wall (red region). It means that if there is any increase of the pressure, the density will also increase or vice versa. The close relationship between the pressure and Mach number is determined from the shock condition, equation (5.26). In addition, a comparison between the one and two-dimensional shock tube test cases can be made when the Mach numbers are close. Therefore, Case 1 for which the Mach number is 1.7 for Channel 1 is a convenient choice since the Mach number was 1.65 for the one-dimensional test case. The acquired physical features i.e., shock waves, contact discontinuity, reflected waves, are also observed from the numerical calculations for the two-dimensional case as in the one-dimensional case. However, the main difference between these two cases is the interaction of the shock with inner and outer walls. For instance, the density is monotonically decreasing in the one-dimensional case as seen in Figure 4.4a, whereas in two-dimensional test case the shock attenuates at the inner wall and gets stronger at the outer wall, as seen in Figure 5.10.

In the present work, the inner and outer Mach numbers at the walls were also computed. It should be first highlighted that this work solely focuses on the evolution of the flow in the curved region while it neglects that in the following rectangular region

as in the experimental work [42]. This can be explicitly seen in Figures 5.12 and 5.13 (also in Figures 5.19 and 5.20) where the numerical solutions of the inner and outer wall Mach numbers are not available for the rectangular region while they exist for the experimental data provided. Although the propagation of the shock wave through the remaining rectangular part of the channel is not in the scope of this work, it can be discussed in a future work.

In Figures 5.12 and 5.13 the comparison of the inner and outer wall Mach numbers between the numerical and the experimental results are depicted for three different grid sizes, 100x400, 200x800 and 400x1600. The grid refinement is increased by halving the previous grid sizes in both the r and θ directions to obtain more accurate agreement with the experimental results. In Figures 5.12 and 5.13 the red and blue shapes represent the numerical and experimental values, respectively. In Figure 5.12, the best agreement is observed from the numerical calculations with 400x1600 computational volumes, due to the greater mesh refinement. However, some of the numerical solutions repeat themselves couple times as in Figure 5.12a. The reason is that the numerical system probably captures the same value of $\max|\partial\rho/\partial\theta|$ a number of times at the traveled distance along the wall and reports the same values. To prevent this, a weighted arithmetic mean average is applied for the angle of the shock wave at the wall to obtain numerical solutions without repetitions. The weighted average for the angle of the shock wave at the wall is determined as

$$\theta_{sh-w} = \frac{\sum_{i=0}^{4} w_{i-2}\theta_{i-2}}{\sum_{i=0}^{4} w_{i-2}}, \tag{5.34}$$

where θ_{sh-w} is the weighted position angle of the shock wave, $w_i = |\partial\rho/\partial\theta|_i$ is the weight, and θ_i is the position angle of the shock wave. The results obtained using equation 5.34 are presented in Figure 5.13. To summarize, the shock has a higher Mach number at the outer wall, $M_w/M_0 > 1$, and the values at the inner wall attenuate gradually with time, $M_w/M_0 < 1$. These results for the wall Mach numbers show better agreement with the experimental data for the increasing outer wall Mach number than for the decreasing inner wall Mach number.

Case 2 : $\mathbf{M_0 = 2.1}$

The second numerical simulations were for the same sharp bend with a stronger initial Mach number, $M_0 = 2.1$. The numerical calculations were run until the final time, $T_{end} = 0.00018\ s$, and the coefficients of the artificial diffusion were 0.16, 0.08 and 0.03 for the three different computational mesh volumes, 100x400, 200x800 and 400x1600, respectively.

In Figures 5.14 and 5.15 the ratios of the wall Mach numbers to initial Mach number along the inner and outer walls are displayed. The traveled distance of the shock wave along the walls is nondimensionalized to have consistency with the experimental work [42]. The wall Mach number increases to a ratio of 1.3 at the outer wall, and the numerical results are in a very good agreement with the experimental results. On the other hand, the ratio of the inner wall shock Mach number decreases to 0.7, a value that is slightly higher than the reported experimental result.

Case 3 : $\mathbf{M_0 = 2.7}$

The last solution for Channel 1 is for the initial Mach number $M_0 = 2.7$. The numerical simulations were computed with the final time, $T_{end} = 0.0001548\ s$. The artificial diffusion coefficients were 0.2, 0.1 and 0.035 for the same three computational mesh domains, 100x400, 200x800 and 400x1600, respectively.

In Figure 5.16, the physical features of the flow field behind the initial shock front, the incident and reflected shocks the Mach stem and the contact discontinuity are clearer than those for the lower initial Mach number. The Mach stem moves normal to the outer wall towards the end of channel. The yellow reflected waves, which bounce from the inner and outer walls, meet in the middle of the tube and continue moving towards the exit of the channel. In addition, the shock wave weakens along the inner wall. The reason is that the expansion waves, which are created along the inner wall, interact with the accelerated triple point against the outer wall. All these physical features are captured accurately using the numerical scheme, as illustrated in Figure 5.2.

Moreover, the amount of compressed gas increases with the initial shock strength. This can be seen from the red regions for the density and pressure in Figures 5.17 and

5.18, respectively. It is clear that there is more compression for the higher initial Mach number as observed by comparing the results for $M_0 = 1.7$ and $M_0 = 2.7$.

The normalized inner and outer shock wall Mach numbers along the nondimensionalized distance were calculated numerically and compared with the experimental data as shown in Figures 5.19 and 5.20. Good agreement is obtained for the Mach numbers at the outer and inner walls for 200x800 computational mesh domain for the weighted results. The ratio of the concave wall Mach number increased to 1.3, while the convex wall Mach numbers decreased to 0.6. Again, the numerical results are closer to the experimental results for the outer wall than for the inner wall, for the same reason is for $M_0 = 1.7$.

Next, numerical results are presented for Channel 2 for three different initial Mach numbers, $M_0 = 1.2,\ 1.9$ and 2.9. Here, the numerical solution and the values of the wall Mach numbers of the shock are also examined numerically as for Channel 1.

Case 1 : $\mathbf{M_0 = 1.2}$

The same procedure as for Channel 1 is followed for the numerical computations for the shallow bend for an initial Mach number $M_0 = 1.2$. Good numerical results are obtained by using adequate enough artificial diffusion constants 0.07, 0.03 and 0.018 for the computational meshes of 100x400, 200x800 and 400x1600, respectively, as for Channel 1. Since the inner and outer radii of the tube are greater in the case of the shallow bend compared with those of the sharp bend, the computations were run for the longer time periods $T_{end} = 0.0005949\ s$.

In Figures 5.21, 5.22 and 5.23 the velocity in the θ direction, the density and the pressure are showed by contour plots. The waves created in the flow field are similar to those observed for Channel 1. However, the same behavior of preserving the initial curvature from the beginning of the channel does not occur. In addition, the shock front is deformed more like a planar shock wave. The reason is that the interaction between the expansion wave and the triple point appears earlier in Channel 2 compared with Channel 1.

In Figures 5.24 and 5.25, the non-weighted and weighted results for the Mach numbers are compared with the inner and outer walls experimental values [42]. For the shallow bend, the outer wall Mach number increases up to around $1.2M_0$ and the inner

wall Mach number decreases to around $0.9M_0$, which is slightly higher than the experimental results. The results obtained using the mesh size of 400x1600 provides the best results, as expected. It should be noted that $0.9M_0 = 1.08$, so that the shock at the inner wall has become weak and near sonic.

Case 2 : $\mathbf{M_0 = 1.9}$

In Case 2 the initial Mach number is increased to 1.9 for the shallow bend. The final time was chosen to be $T_{end} = 0.000468\ s$, and artificial viscosity coefficients to eliminate the non-physical numerical oscillations were 0.13, 0.065 and 0.035 for all of the selected computational mesh sizes from large to small, respectively.

In Figures 5.26 and 5.27, the Mach numbers for inner and outer walls are shown. It is observed that the calculated results match better with the experimental results for the outer wall, as for Channel 1. The numerical results decreases to around $0.8M_0$ for the inner wall Mach number, while the experimental ones are less than $0.7M_0$. This may be due to the damping effect of the artificial viscosity.

Case 3 : $\mathbf{M_0 = 2.9}$

In Case 3 the initial shock wave has the highest Mach number $M_0 = 2.9$. The final time, $T_{end} = 0.0003636\ s$, is lower in this case than the other two cases due to speed of the strong shock. The coefficients of artificial diffusion are 0.26, 0.13 and 0.068 for the three mesh volumes 100x400, 200x800 and 400x1600, respectively.

In Figures 5.28, 5.29 and 5.30 contour plots for the velocity in θ direction u_θ, the density and the pressure are shown, respectively. The numerical velocity indicates that the Mach stem hit the inner wall before end of the curvature of the channel and this cause shock wave becoming straight as at the beginning of the channel. The main reason is the interaction between the traveling Mach triple point and the expansion wave. This can also be seen from the numerical density, as sketched in Figure 5.29. The red area of compressed gas gradually increases and moves towards the inner wall.

In Figures 5.31 and 5.32, while the outer wall Mach numbers start to decrease at the end of the tube, the inner wall Mach numbers reach a plateau. This behavior is in agreement with the experimental data. This is because when the Mach triple point reaches the inner wall, the shock at the outer wall loses its strength, while the shock wave at the inner wall strengthens. Again this is excellent agreement between

the Mach number at the inner and outer walls as given by the numerical solutions and experimental results. However, it should be pointed out that the difference between numerical solution and experimental data is more significant for the inner wall Mach numbers compared to the outer wall Mach numbers. The main reason for that is the value of the artificial viscosity which is higher than that of the real viscosity.

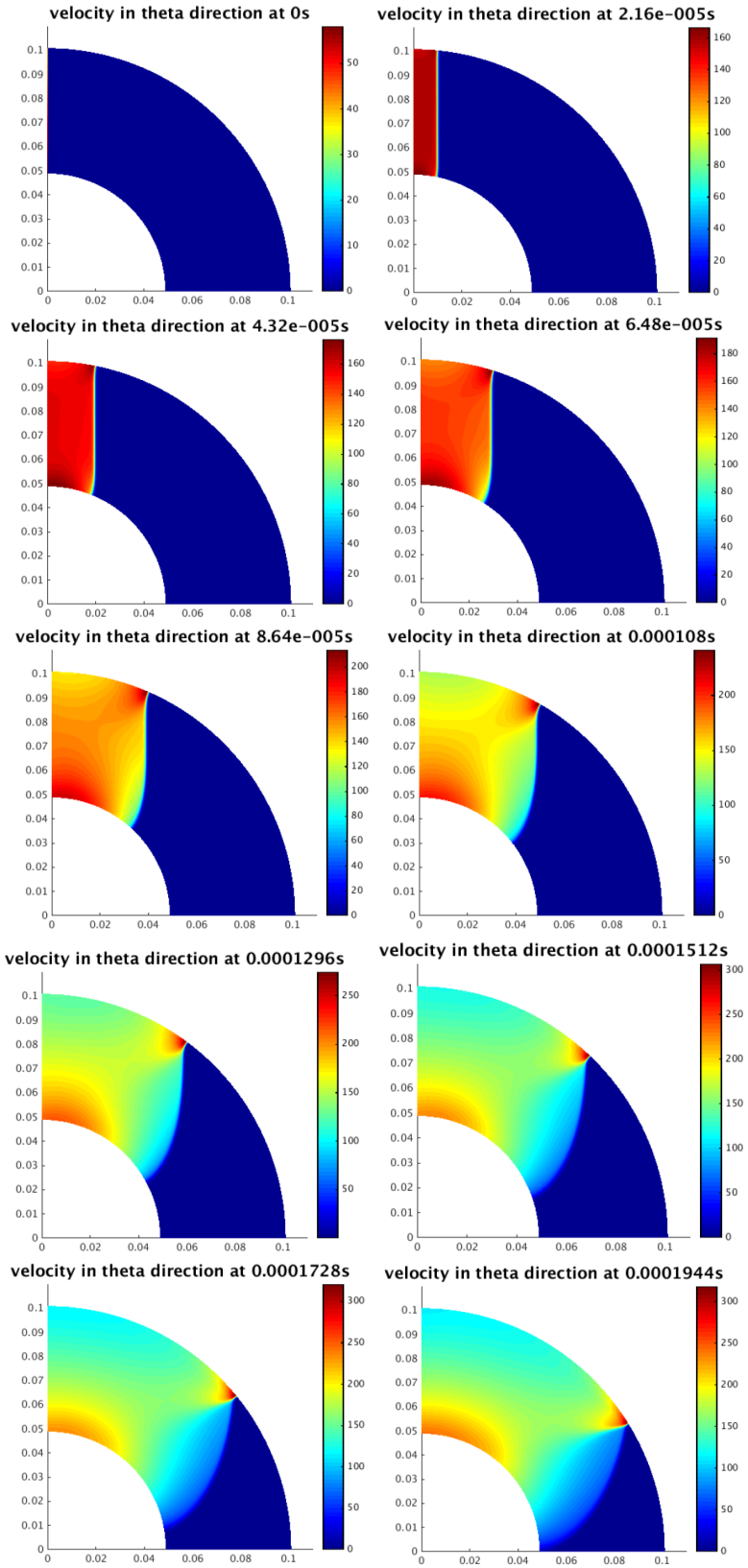


Figure 5.9: Numerical evolution of θ velocity u_θ for initial Mach number, $M_0 = 1.7$. (channel 1)

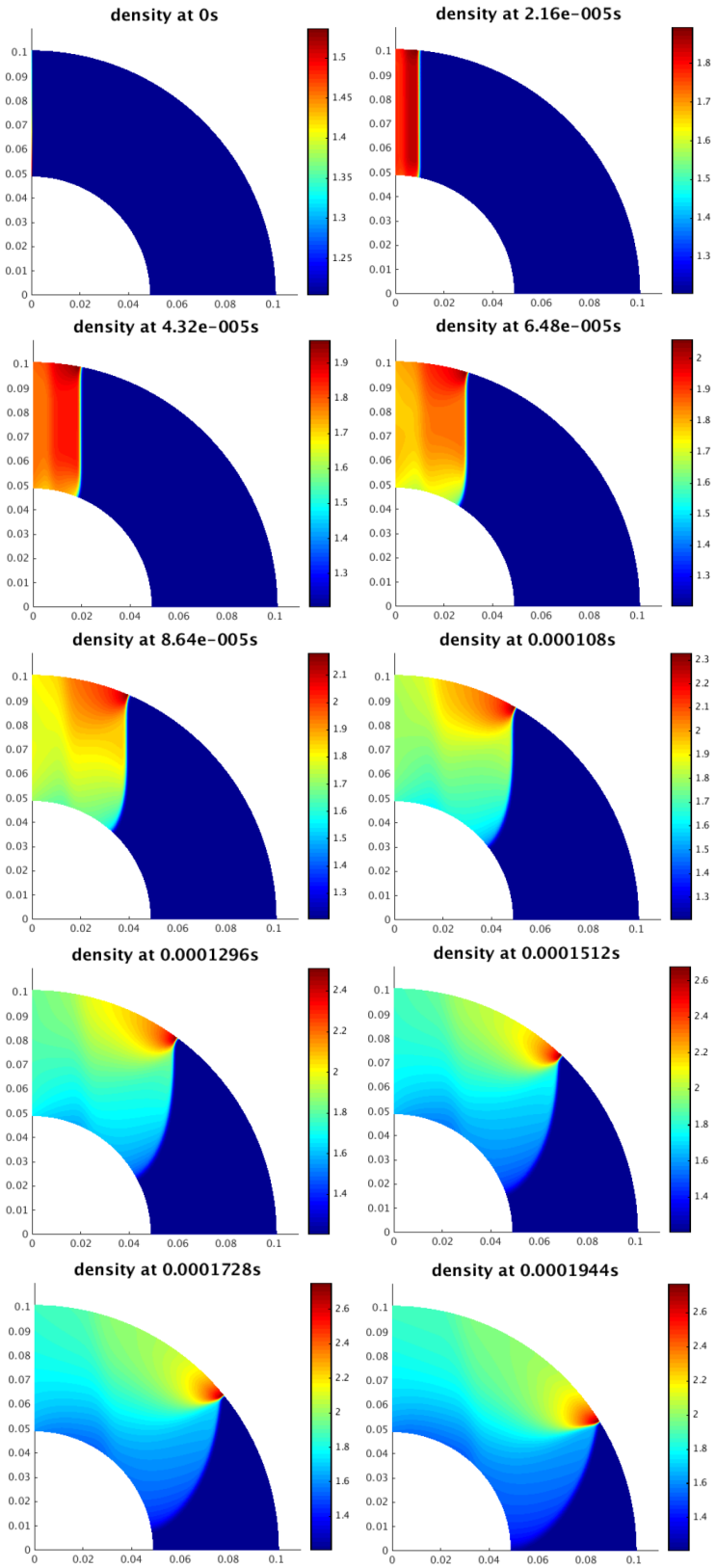


Figure 5.10: Numerical evolution of density for initial Mach number, $M_0 = 1.7$. (channel 1)

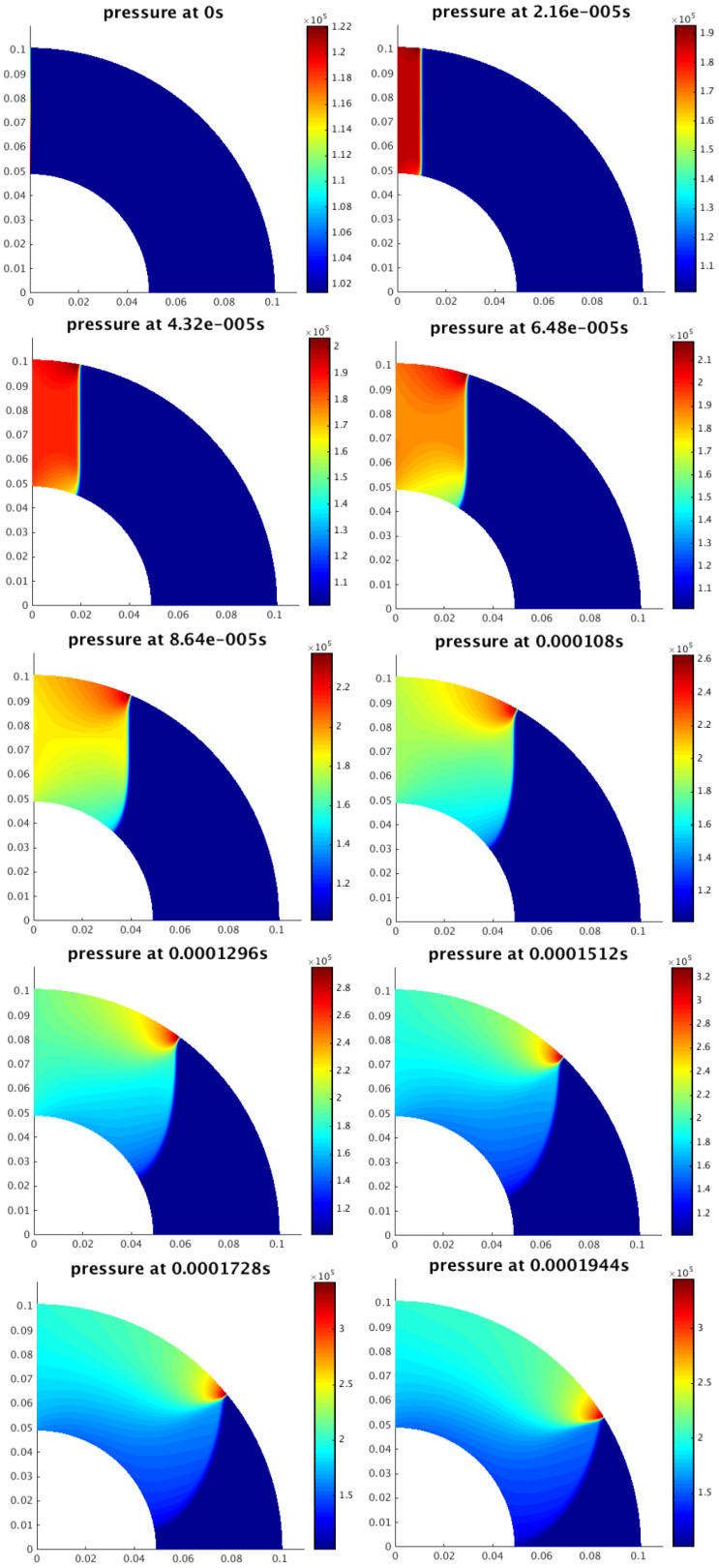
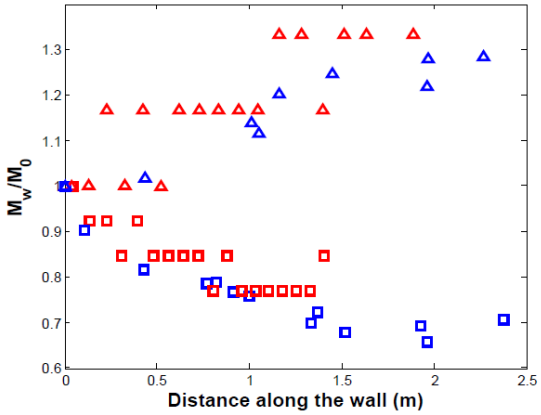
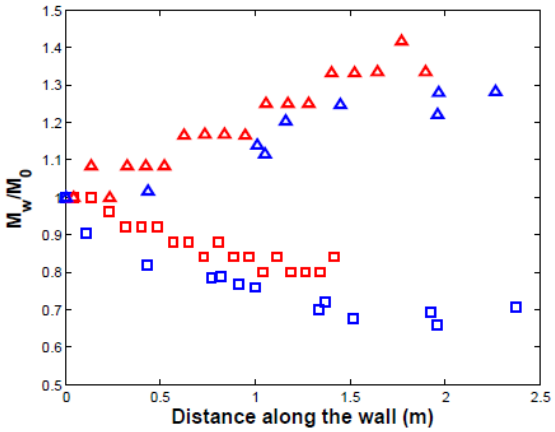


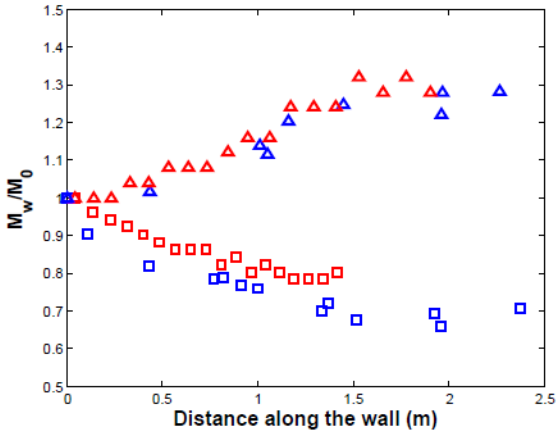
Figure 5.11: Numerical evolution of pressure for initial Mach number, $M_0 = 1.7$. (channel 1)



(a) Results for 100x400 grid cells

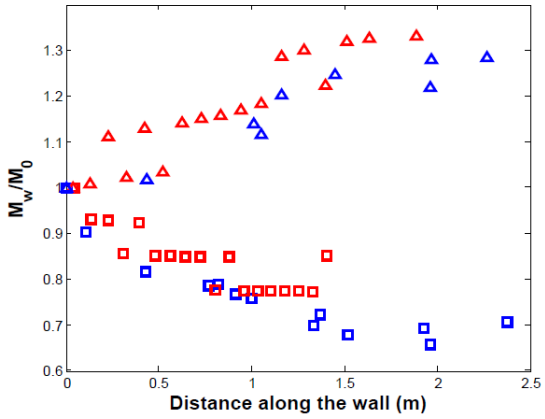


(b) Results for 200x800 grid cells

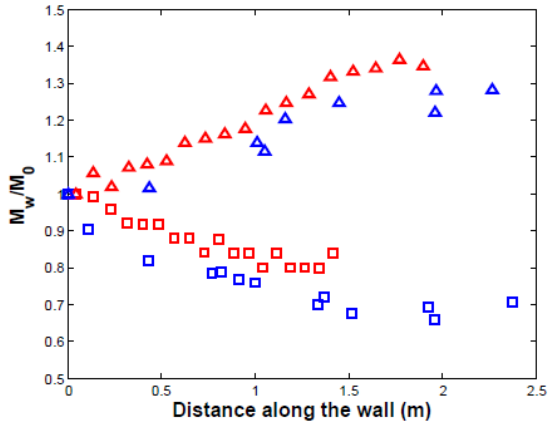


(c) Results for 400x1600 grid cells

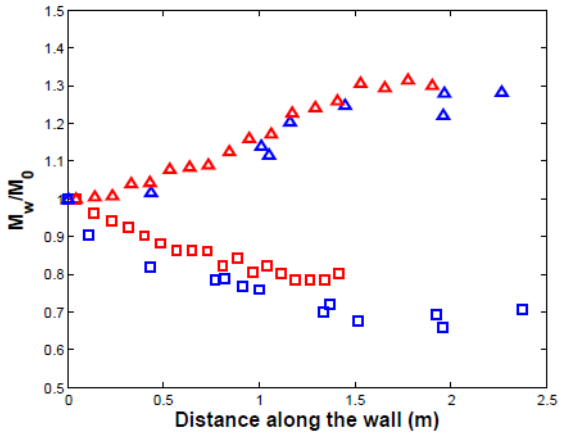
Figure 5.12: Initial Mach number 1.7 for channel 1. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively.



(a) Weighted results for 100x400 grid cells

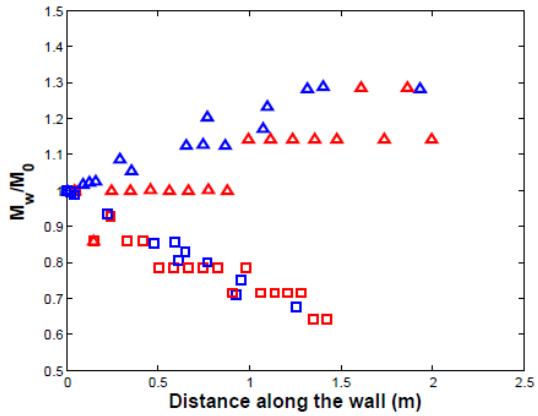


(b) Weighted results for 200x800 grid cells

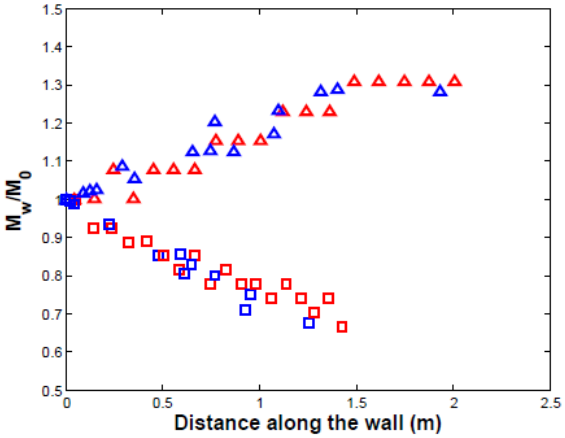


(c) Weighted results for 400x1600 grid cells

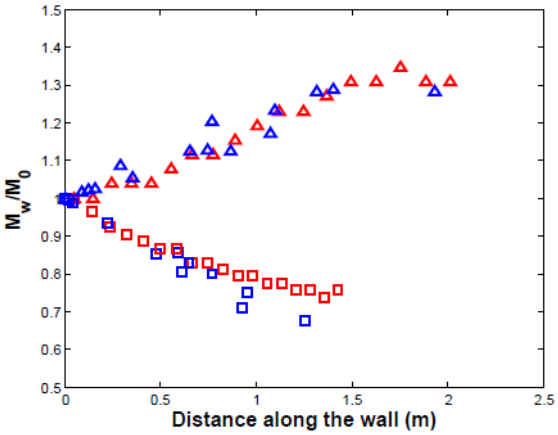
Figure 5.13: Initial Mach number 1.7 for channel 1. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively.



(a) Results for 100x400 grid cells

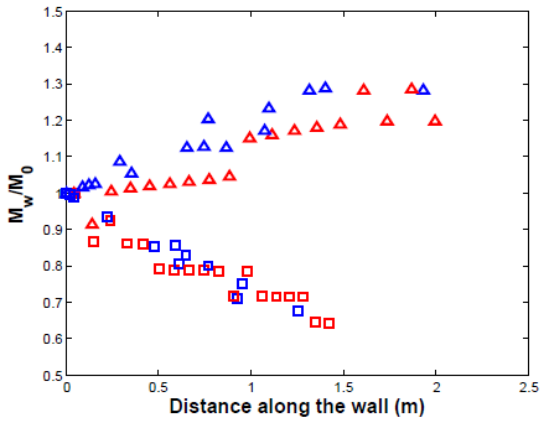


(b) Results for 200x800 grid cells

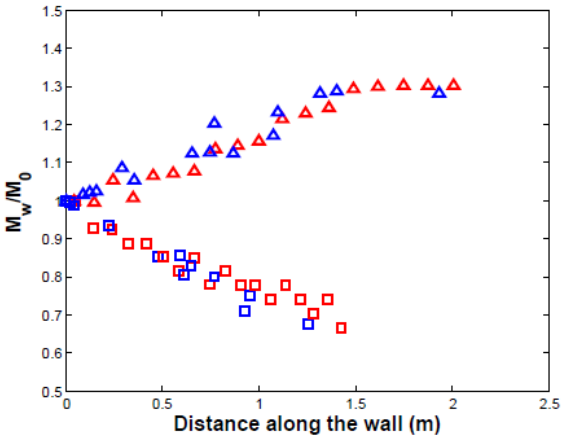


(c) Results for 400x1600 grid cells

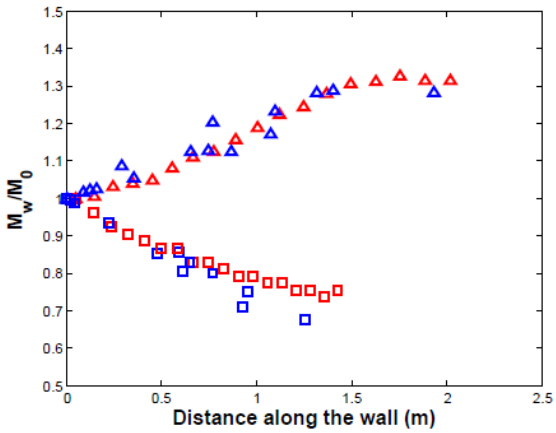
Figure 5.14: Initial Mach number 2.1 for channel 1. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively.



(a) Weighted results for 100x400 grid cells



(b) Weighted results for 200x800 grid cells



(c) Weighted results for 400x1600 grid cells

Figure 5.15: Initial Mach number 2.1 for channel 1. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively.

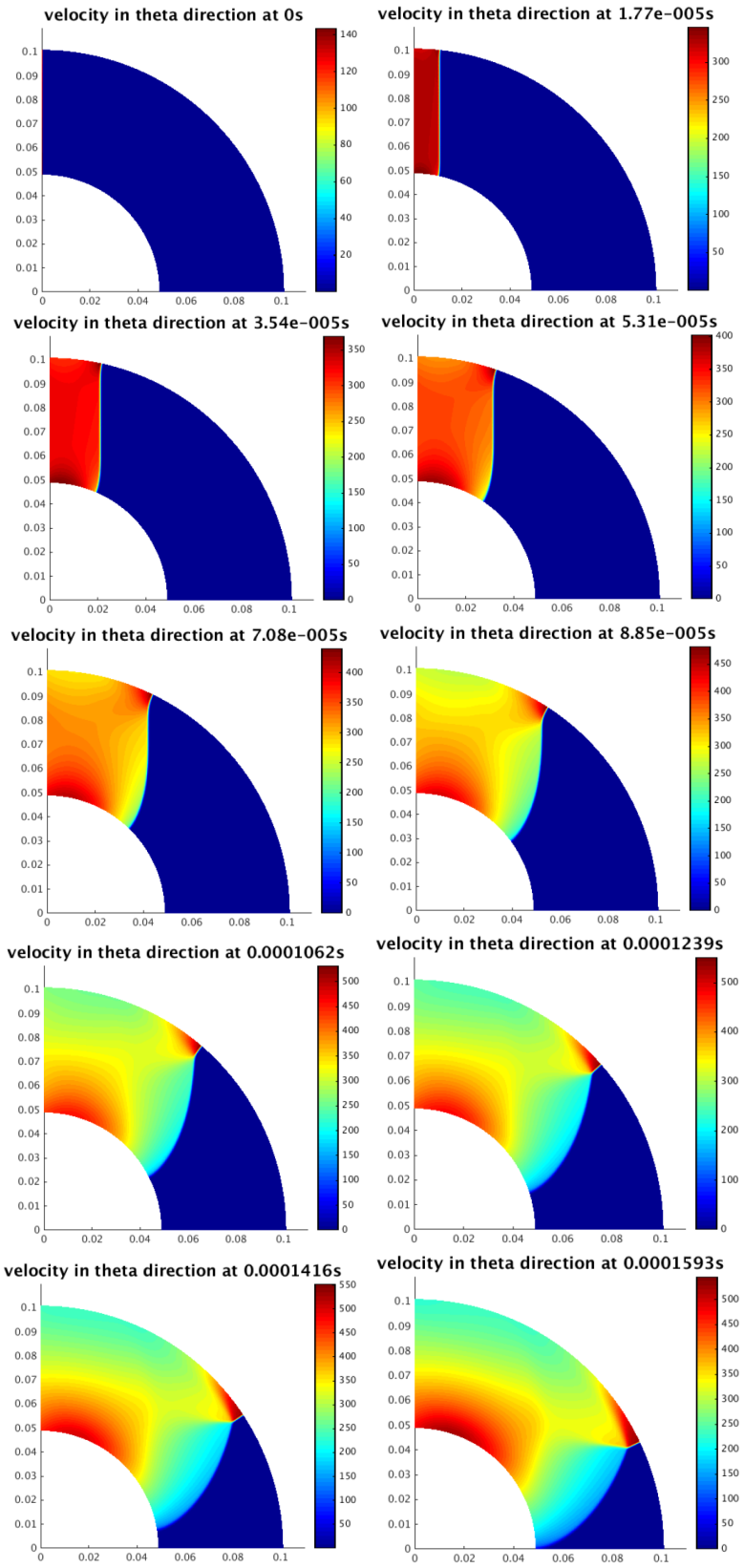


Figure 5.16: Numerical evolution of θ velocity u_θ for initial Mach number, $M_0 = 2.7$. (channel 1)

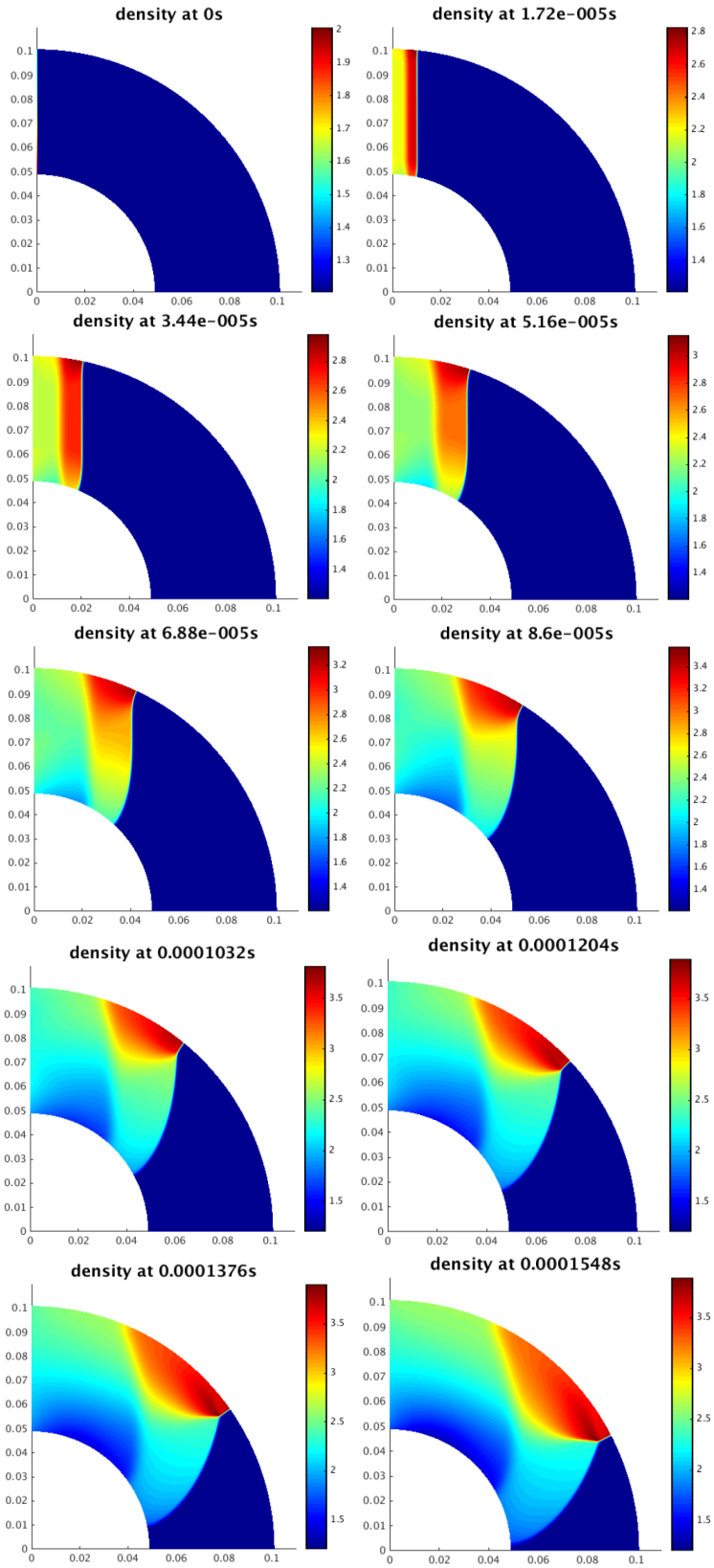


Figure 5.17: Numerical evolution of density for initial Mach number, $M_0 = 2.7$. (channel 1)

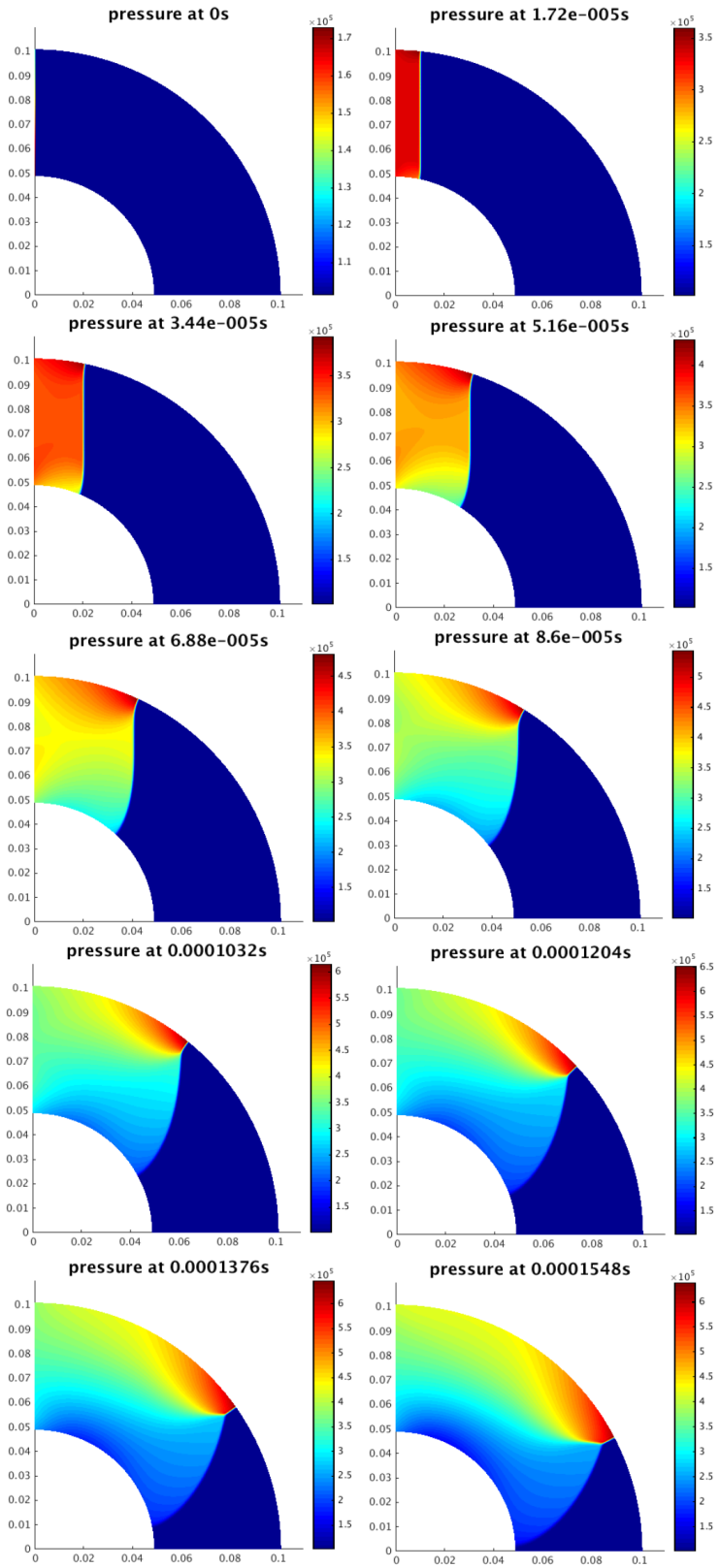
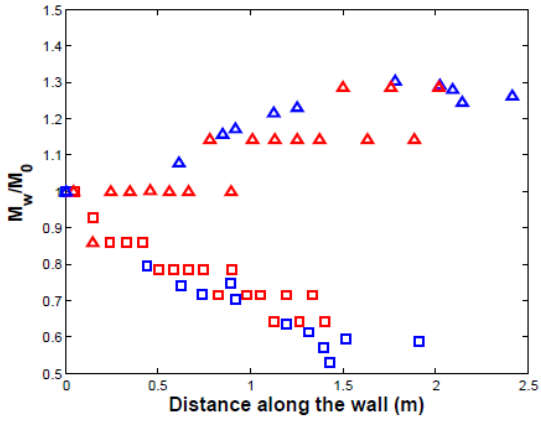
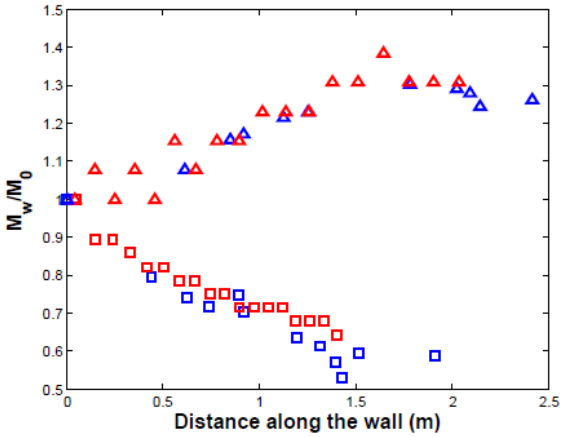


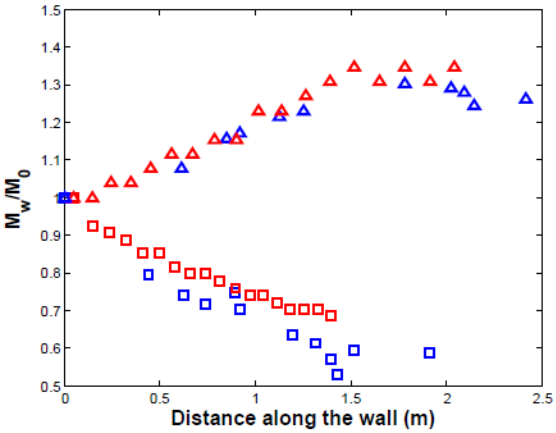
Figure 5.18: Numerical evolution of pressure for initial Mach number, $M_0 = 2.7$. (channel 1)



(a) Results for 100x400 grid cells

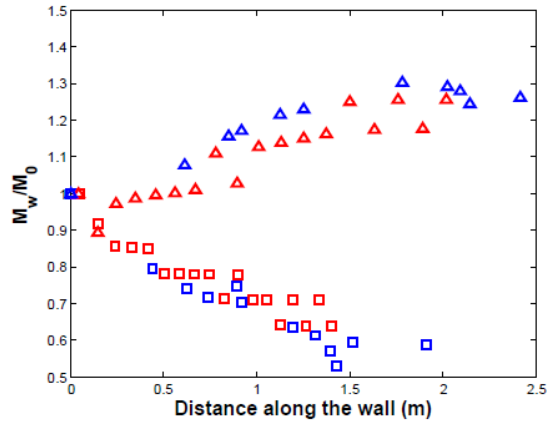


(b) Results for 200x800 grid cells

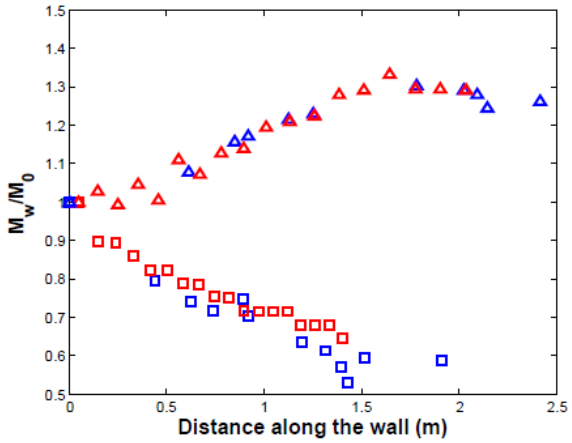


(c) Results for 400x1600 grid cells

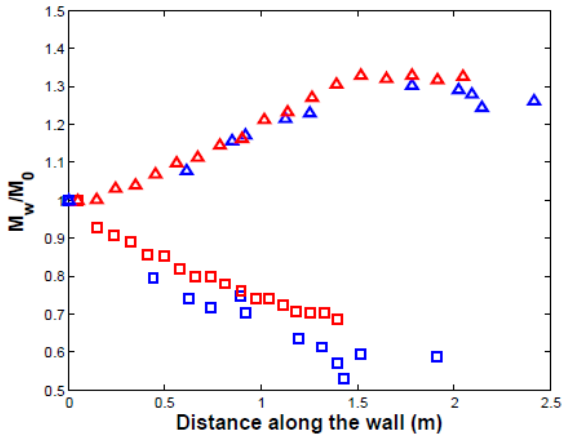
Figure 5.19: Initial Mach number 2.7 for channel 1. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively.



(a) Weighted results for 100x400 grid cells



(b) Weighted results for 200x800 grid cells



(c) Weighted results for 400x1600 grid cells

Figure 5.20: Initial Mach number 2.7 for channel 1. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively.

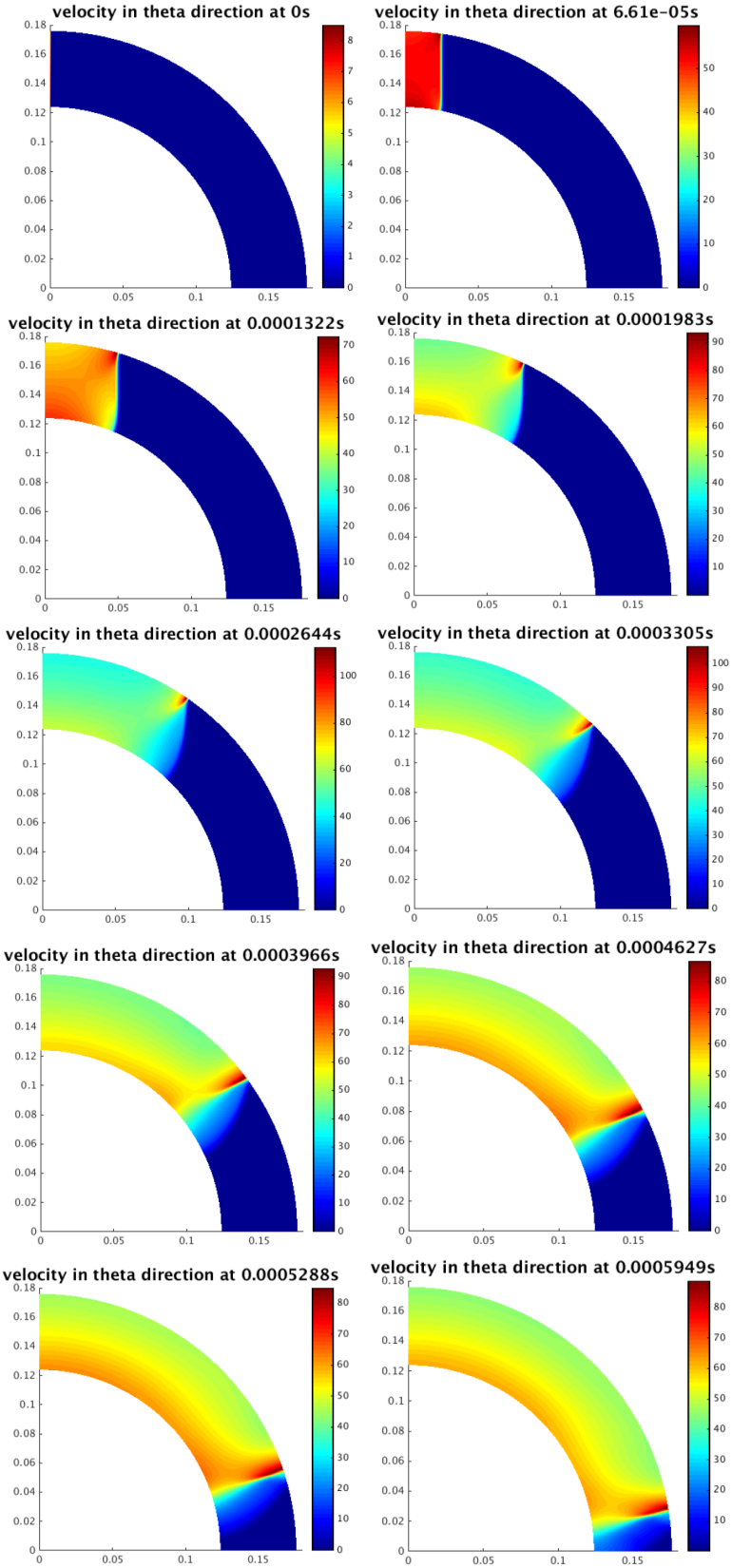


Figure 5.21: Numerical evolution of the θ velocity u_θ for initial Mach number, $M_0 = 1.2$. (channel 2)

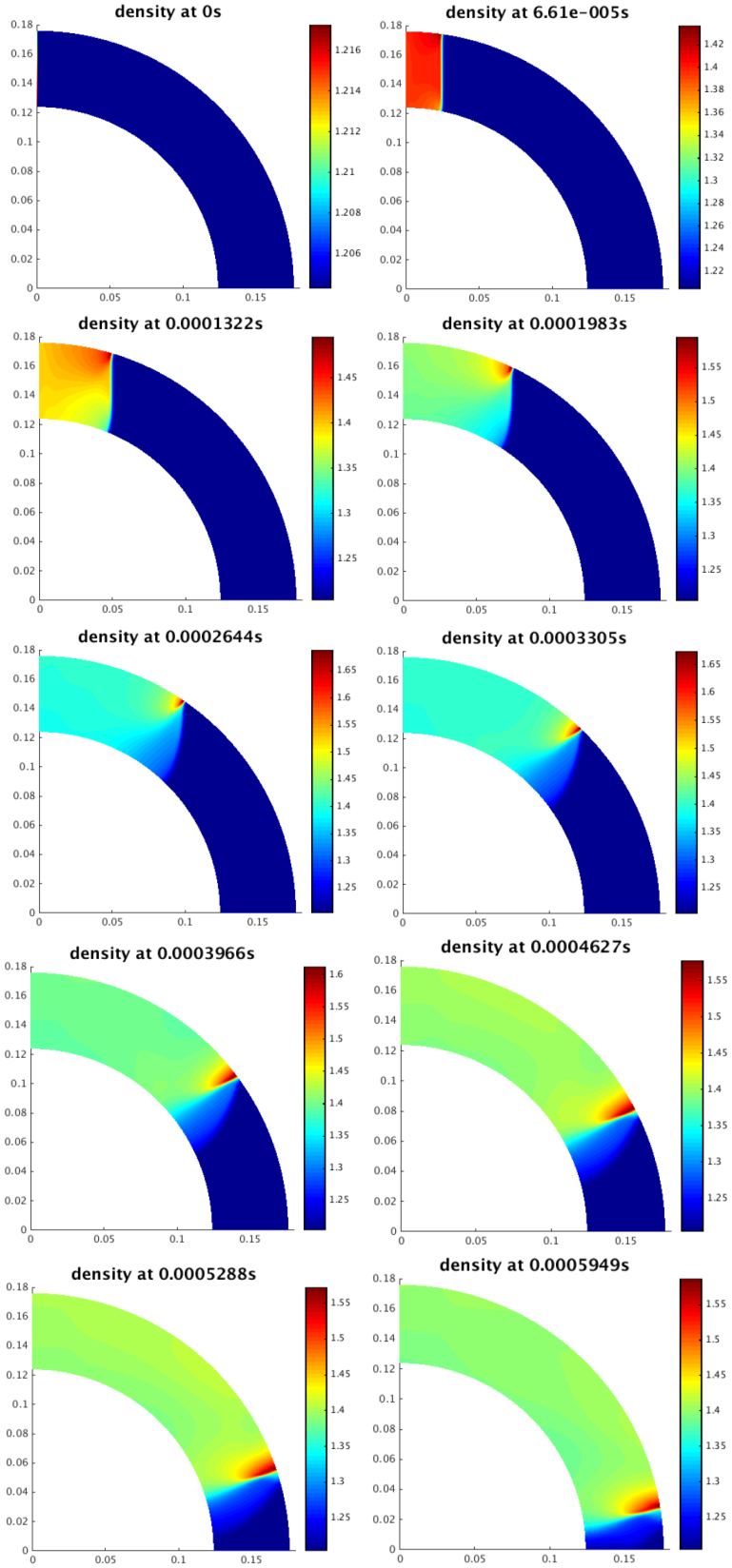


Figure 5.22: Numerical evolution of density for initial Mach number, $M_0 = 1.2$. (channel 2)

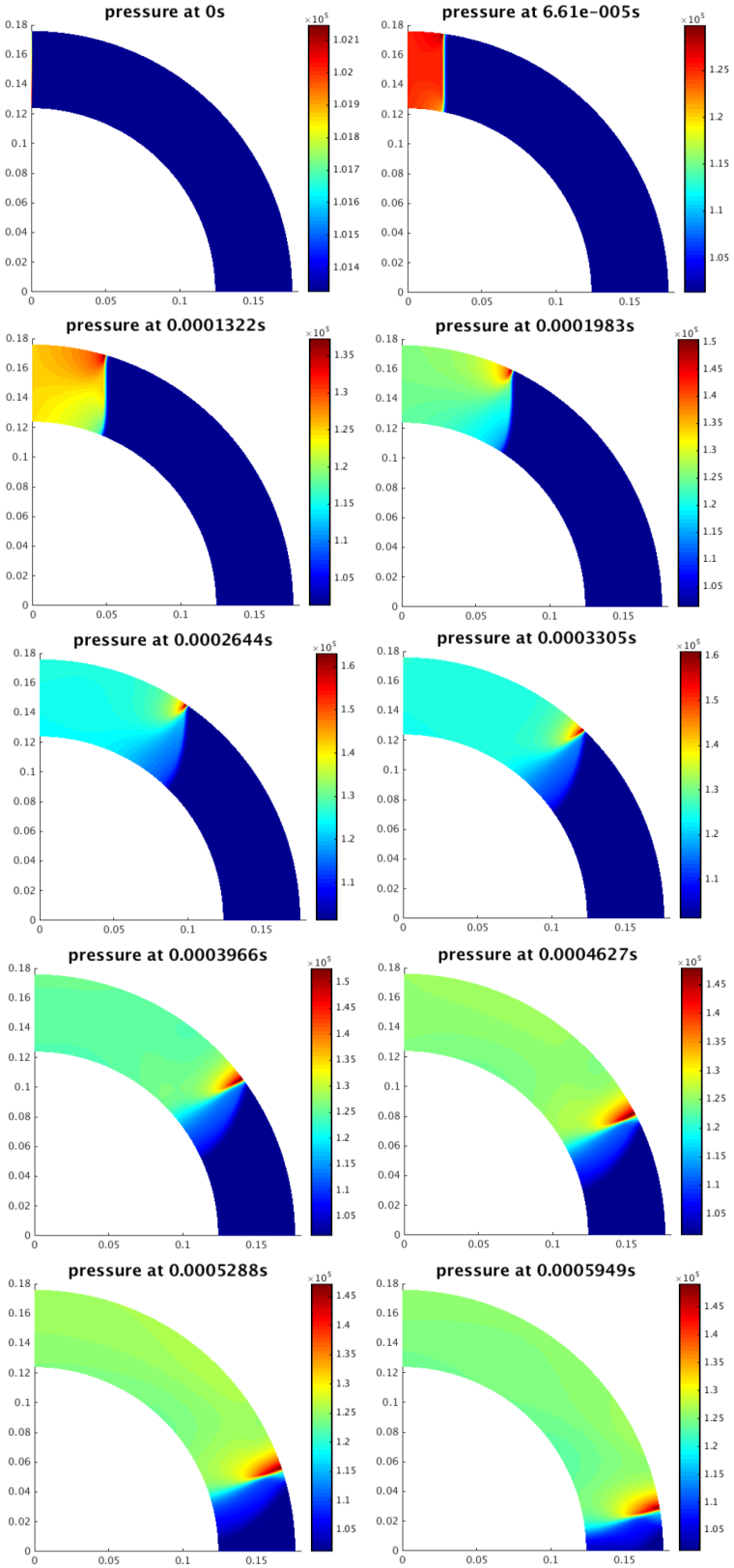
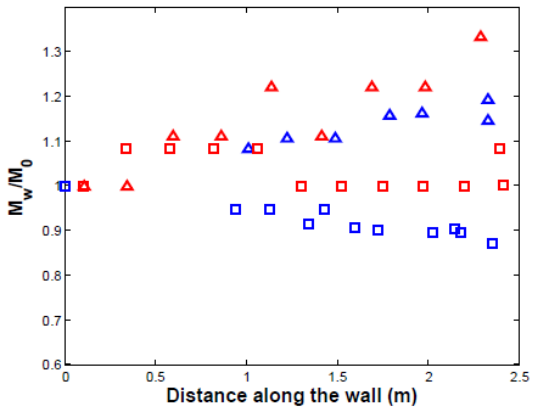
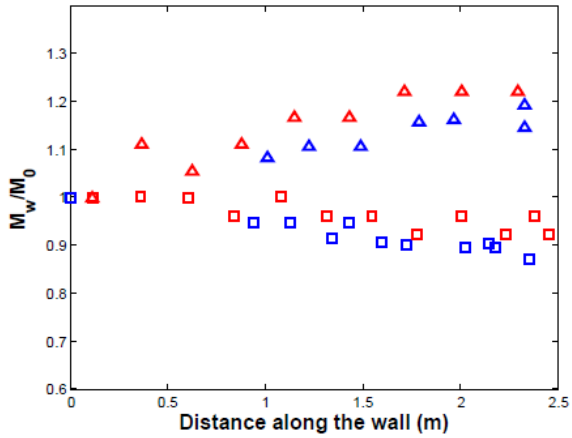


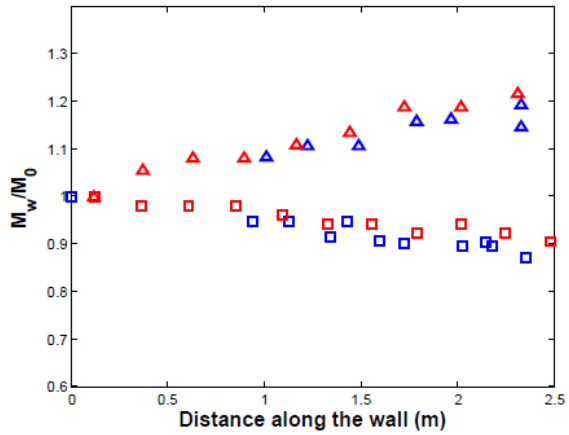
Figure 5.23: Numerical evolution of pressure for initial Mach number, $M_0 = 1.2$. (channel 2)



(a) Results for 100x400 grid cells

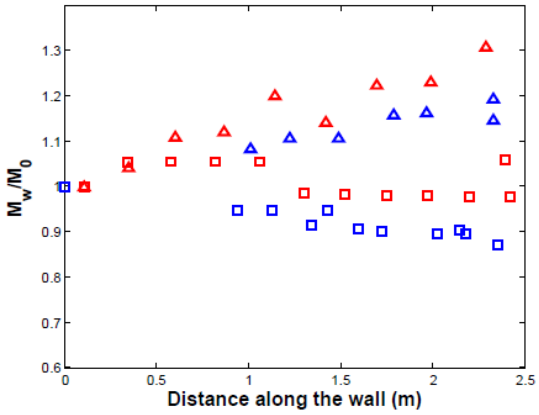


(b) Results for 200x800 grid cells

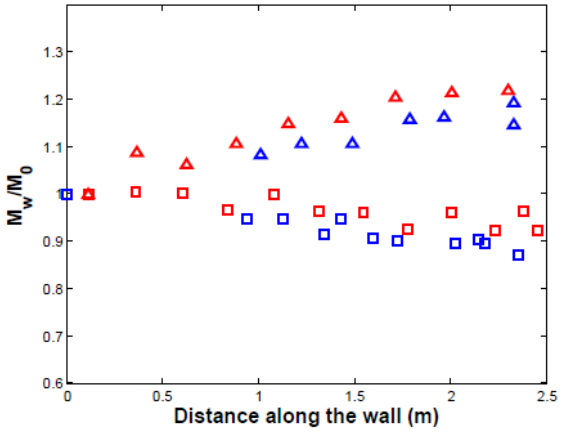


(c) Results for 400x1600 grid cells

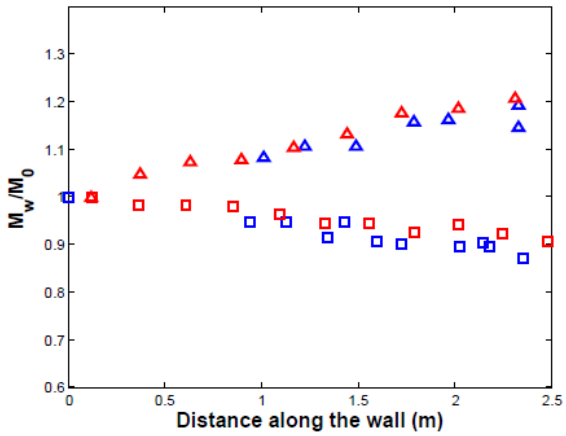
Figure 5.24: Initial Mach number 1.2 for channel 2. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively.



(a) Weighted results for 100x400 grid cells

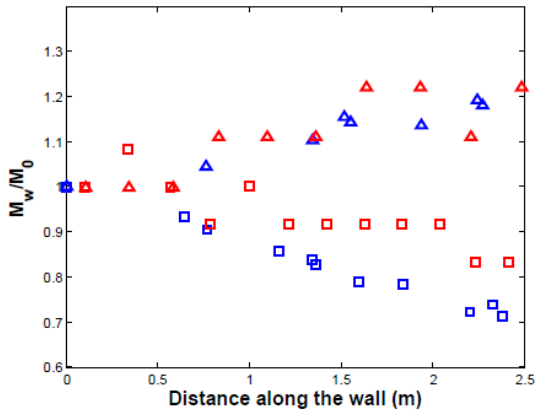


(b) Weighted results for 200x800 grid cells

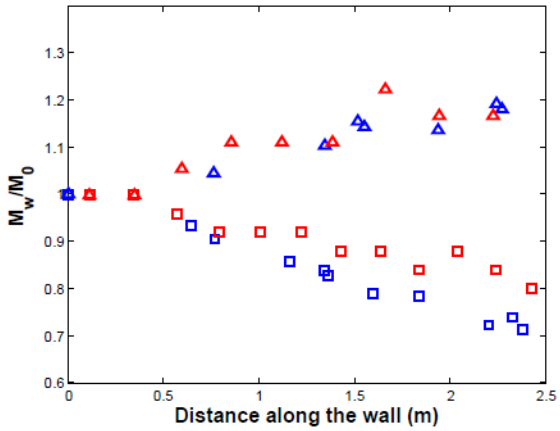


(c) Weighted results for 400x1600 grid cells

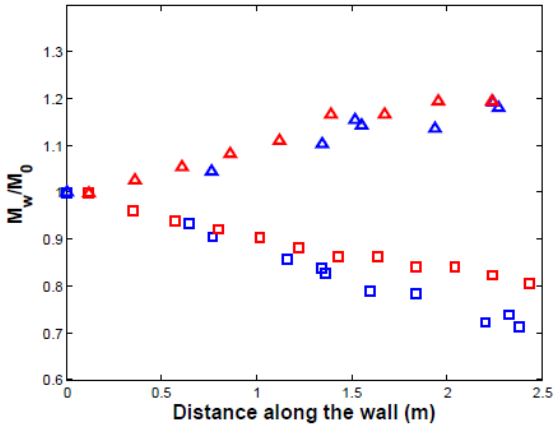
Figure 5.25: Initial Mach number 1.2 for channel 2. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively.



(a) Results for 100x400 grid cells

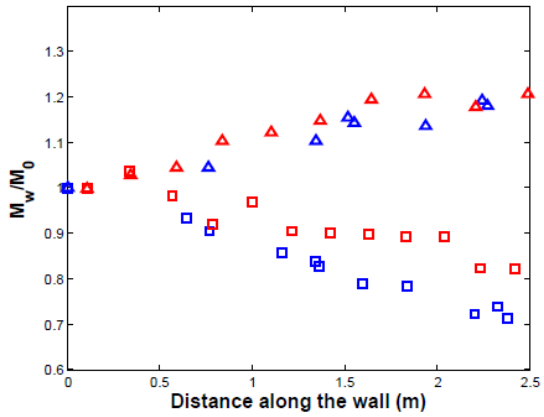


(b) Results for 200x800 grid cells

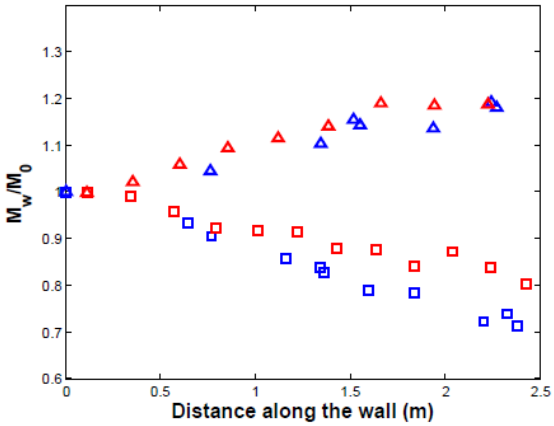


(c) Results for 400x1600 grid cells

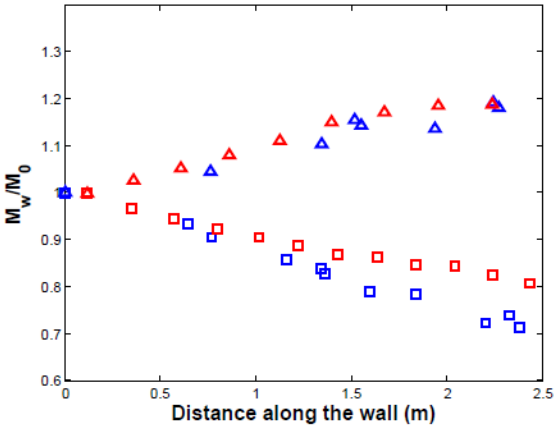
Figure 5.26: Initial Mach number 1.9 for channel 2. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively.



(a) Weighted results for 100x400 grid cells



(b) Weighted results for 200x800 grid cells



(c) Weighted results for 400x1600 grid cells

Figure 5.27: Initial Mach number 1.9 for channel 2. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively.

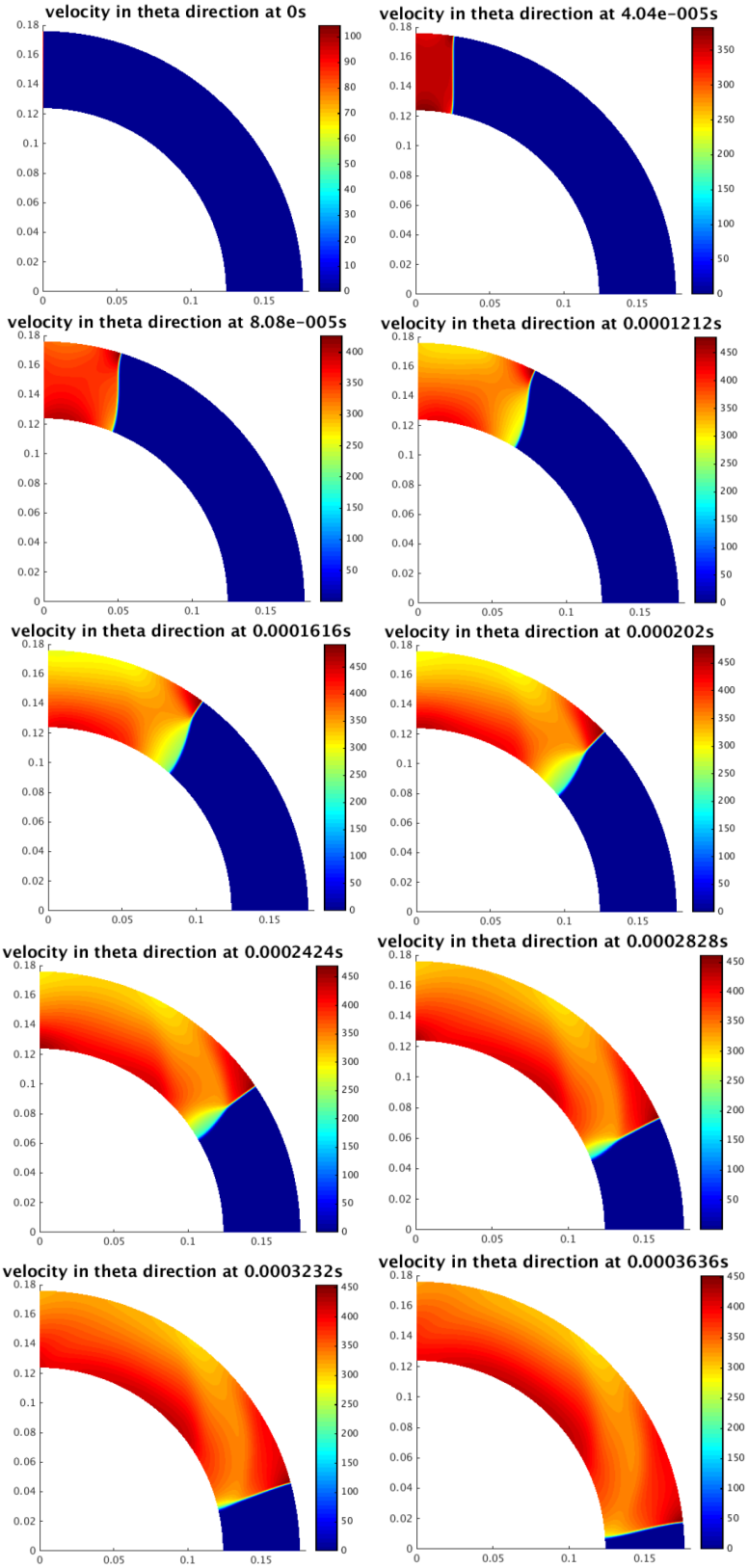


Figure 5.28: Numerical evolution of θ velocity u_θ for initial Mach number, $M_0 = 2.9$. (channel 2)

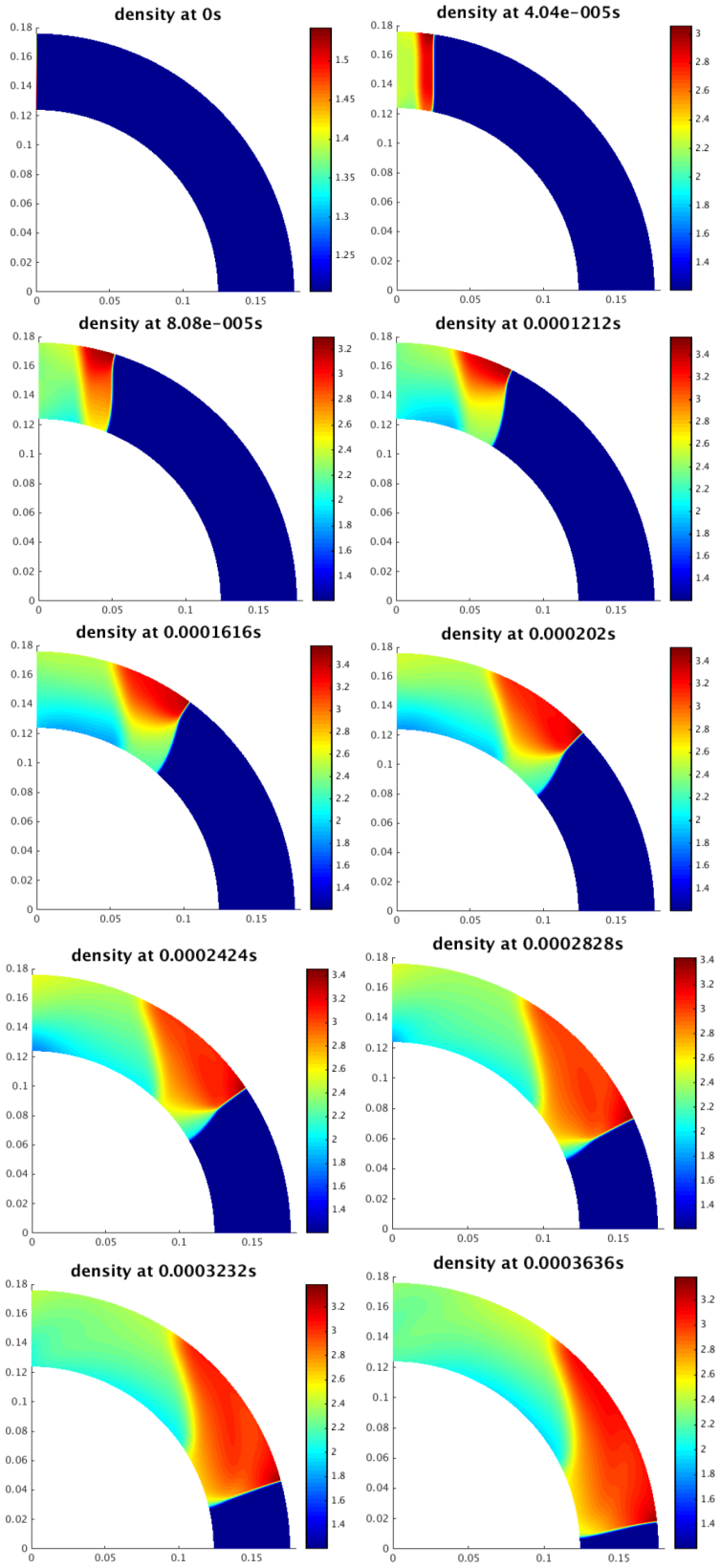


Figure 5.29: Numerical evolution of density for initial Mach number, $M_0 = 2.9$. (channel 2)

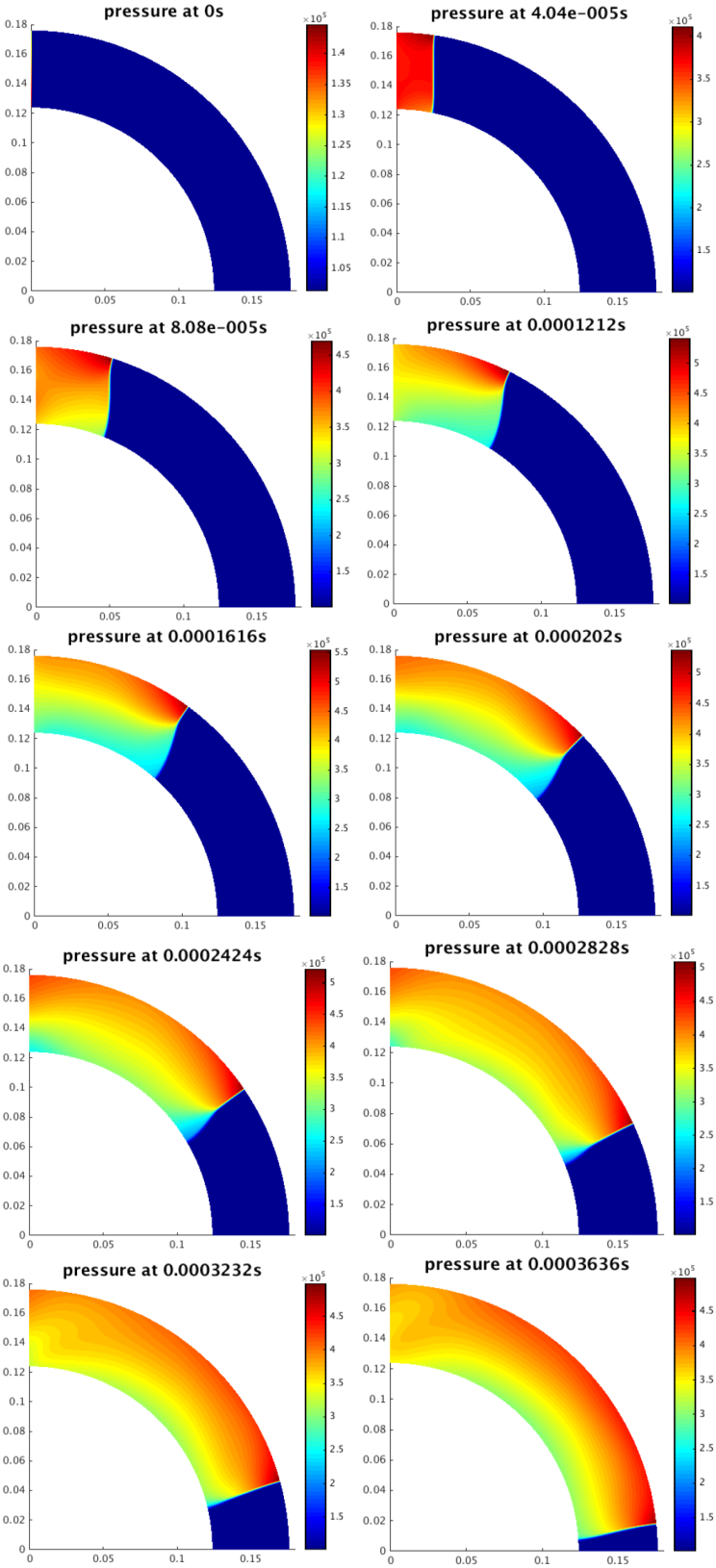
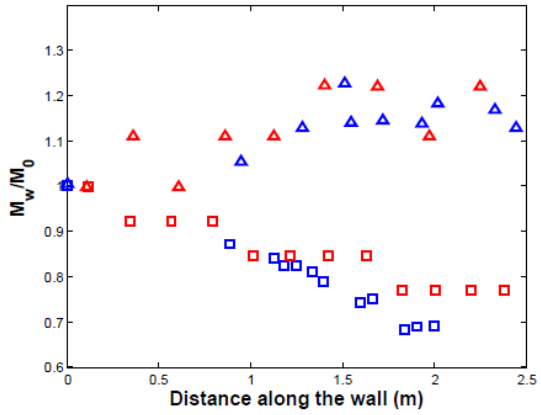
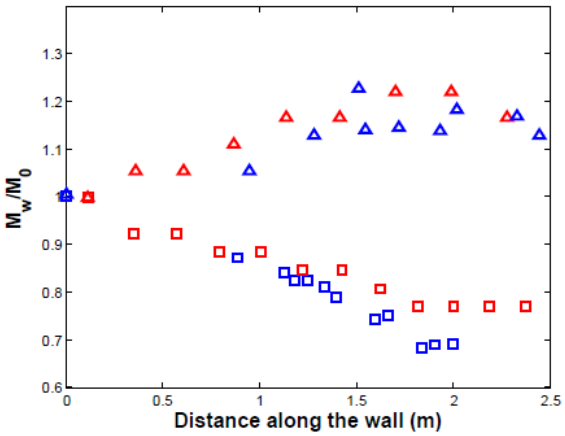


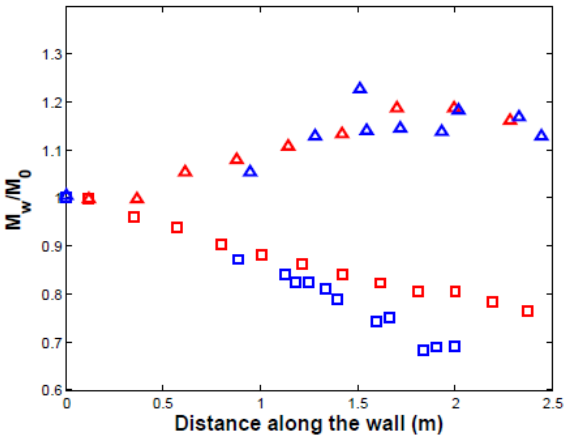
Figure 5.30: Numerical evolution of pressure for initial Mach number, $M_0 = 2.9$. (channel 2)



(a) Results for 100x400 grid cells

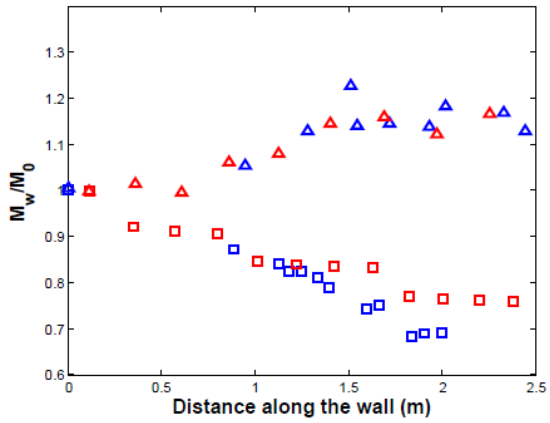


(b) Results for 200x800 grid cells

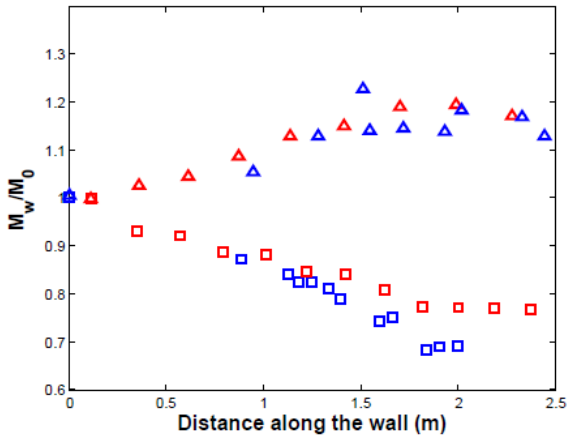


(c) Results for 400x1600 grid cells

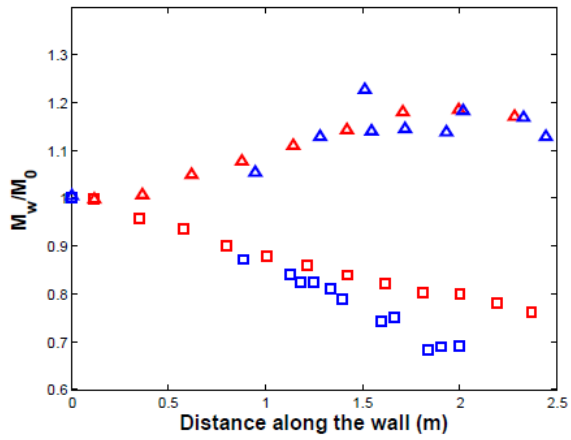
Figure 5.31: Initial Mach number 2.9 for channel 2. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively.



(a) Weighted results for 100x400 grid cells



(b) Weighted results for 200x800 grid cells



(c) Weighted results for 400x1600 grid cells

Figure 5.32: Initial Mach number 2.9 for channel 2. Blue and red triangles represent experimental data and numerical results on the outer wall, respectively. Blue and red squares represent experimental data and numerical results on the inner wall, respectively.

Chapter 6

Conclusions and Future Work

In this dissertation, numerical schemes to solve the system of non-linear hyperbolic conservation laws governing gas flow in one and two dimensional spatial domains have been derived. Two main objectives were presented. In Chapter 2, entropy stable boundary schemes were derived for the one dimensional Euler equations in cartesian coordinate, subject to physical boundary conditions, i.e., grid interface, far-field and no-penetration wall boundary conditions [29]. In Chapters 3-5, the construction of the CCFVM method was presented for the two dimensional compressible Euler equations in polar coordinates to study the propagation of shock waves along the inner and outer walls of 90° curved channels. The results were compared with experimental data given in [42].

In Chapter 2, entropy stable boundary schemes were derived to obtain stable numerical solutions of the one-dimensional compressible Euler equations in the presence of boundaries. In Tadmor [24], a specific way to prove the stability of numerical schemes to capture shocks was found. Discontinuous solutions containing shock waves have to be defined in a weak sense (as integrals). However, weak solutions are non-unique. Therefore an entropy stability condition was included to choose the physically relevant weak solution [25]. This condition was first discretized and then implemented into the numerical scheme with sufficient numerical diffusion to smooth the shock. The amount of numerical diffusion was quantified in order to satisfy entropy stability by using a comparison principle between entropy conservative and entropy stable schemes [25]. This leads to a bound on the numerical solution by proving local and global entropy

stability conditions. However, a global entropy estimate was not derived by Tadmor [24] since only initial value problems were considered in his study. These problems satisfy the global entropy estimate automatically. To achieve the entropy stability condition globally when boundaries are present, the further investigations were addressed to obtain accurate entropy stable boundary numerical schemes.

To construct entropy stable boundary schemes, the main tool of convex physical entropy was used to satisfy the numerical entropy condition locally and globally in order to derive a bound on the numerical solution, i.e., stability. Specifically, three different boundary conditions were enforced weakly by satisfying a global entropy stability condition, i.e., the far-field characteristic and no-penetration wall boundary conditions and the grid interface treatment. To do this, the numerical flux functions were derived using Roe's entropy conservative scheme for smooth regions and including Roe average numerical diffusion for non-smooth regions, which guaranties entropy stability. The limiter function was used to monitor the position of the shock wave and to provide a condition to add Roe average numerical diffusion in such regions. Moreover, the fourth order entropy stable boundary scheme in [35] was modified subject to physical entropy by using the boundary scheme of Chapter 2 [29]. Numerical simulations demonstrate the robustness and stability of these second and fourth order entropy stable boundary schemes.

The entropy stable schemes were developed based on the ability of the proposed entropy functions to symmetrize the governing equations. While there are many entropy functions which can be applied to the compressible Euler equations, there is only one entropy function that works for the Navier-Stokes equations, i.e., physical entropy. Therefore, the proposed study can be considered as a stepping stone towards constructing entropy stable boundary schemes for the Navier-Stokes equations.

In Chapter 3, the MMS method was used to test verification of the numerical method for the Euler equations in a two dimensional domain. The advantage of using the MMS method is the ability to verify that the scheme solves the governing equations with the correct order of accuracy. However, it does not provide any information about the robustness, convergence and stability of the scheme. In the MMS method, first, convenient benchmark solutions were selected. These were then used to verify

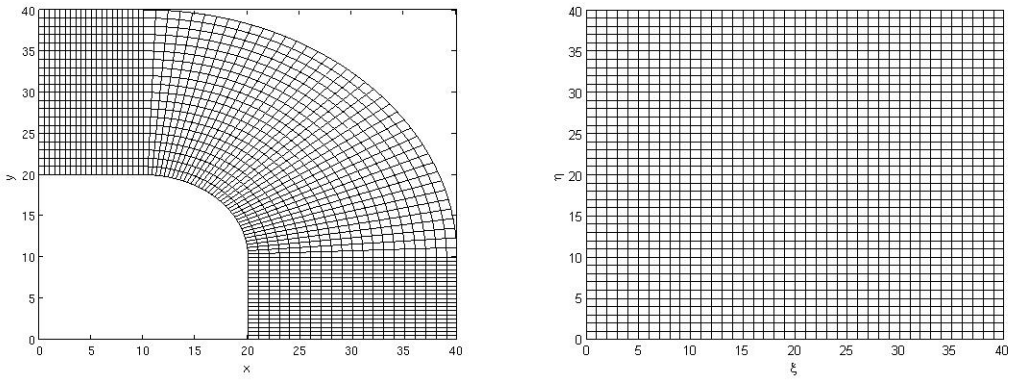
numerical solutions obtained using CCFVM method. Finally, the difference between the benchmark and numerical solutions was calculated in an L^2 norm and was shown to be consistent with second order accuracy for smooth solutions in the interior part of the steady CCFVM. It is worth noting that only the spatial discretization was tested by MMS method since a semi-discretized numerical scheme was applied to the governing equation. The numerical convergence rates were illustrated on a log-log plot for each of the conserved variables.

In Chapter 4, Sod's shock tube problem detailed in [54] was studied numerically to validate the developed CCFVM and SSPRK schemes for the spatial and time discretizations, respectively. These numerical methods were implemented in the discretized, unsteady nonlinear hyperbolic system of compressible Euler equations in a two dimensional domain. Since this experimental test case can only be applied to one-dimensional problems, each dimension was tested individually. Sod's test case is a rigorous and difficult test for numerical schemes since the exact solution includes discontinuities (shock waves); therefore, it is challenging to capture shock waves without generating nonphysical oscillations [61]. The numerical solutions were compared with the exact solution and good agreement was found. The results in each coordinate direction demonstrate that the theoretical accuracy of the proposed numerical scheme was obtained for smooth regions and the shock wave was captured sharply without any nonphysical oscillations. In addition, the contact discontinuity has been smeared slightly due to the numerical diffusion.

In Chapter 5, a previous experimental investigation of the propagation of initially planar shock waves in air around 90° circular sharp and shallow bends by [42] was studied numerically based on the compressible gas equations [42]. The numerical CCFVM and SSPRK methods were implemented in the discretized Euler equations of gas dynamics in polar coordinates, which included artificial diffusion terms. These were augmented by the correct physical initial and boundary conditions. The initial density and pressure were calculated based on the reported initial experimental Mach numbers [10]. In addition, imposed outflow conditions were applied in the numerical scheme to let the flow pass out of the region without serious reflections. No penetration wall boundary conditions were implemented through the use of ghost cells. Numerical results were

obtained to visualize the shock wave motion and the flow behind the shock wave. The numerical scheme was found to provide results which are in excellent agreement with the experimental results. The level of agreement was surprising given that physical effects such as viscosity and possible turbulence have been neglected. A previous study of shock propagation around a 90° bend in [43] based on geometrical shock dynamics also found good agreement with experimental result. However, the present work gives much better agreement with the experimental results than [43] as geometrical shock dynamics is an approximate theory [10]. This is because the flow field around and behind the shock front were not included due to the restrictions of proposed Whitham's theory of geometrical shock dynamics [43].

The physical domain of the experimental study in [42] was originally formed by two rectangular and one curvilinear region, as illustrated in Figure 6.1a. In Figure 6.1b, the computational domain is also illustrated. Since the initially planar shock wave is



(a) Physical Domain **(b)** Computational Domain

Figure 6.1: The physical and computational domains for the experimental study.

located at the beginning of the curved section, the numerical results of Chapter 5 are not affected due to the neglect of the straight section. However, simulations have to be stopped at the end of 90° curved region due to the lack of the experimental straight shock tube at this end. As a future work, the numerical method can be improved by using generalized curvilinear coordinates so that it includes the entire physical experimental domain. The full motion of the shock wave, including in the beginning and terminating straight sections of the shock tube, can then be studied. Also, the present study can be extended by including the entropy stability theory, including limiter func-

tions as described in Chapter 2 for one dimension. Furthermore, the shock strengths at the convex and concave walls are not equal. The size of the artificial diffusion could then be varied along the shock. It is important to note that these entropy stable schemes in curvilinear coordinates were derived by taking into consideration three important ingredients; **(i)** entropy conservative fluxes, **(ii)** numerical diffusion operators which can damp the oscillations around the a shock wave, and **(iii)** metric tensor relations for one to one transformation from the physical domain to the computational domain. The final is important for numerically solving for the supersonic flow in the curved experimental shock tube illustrated in Figure 6.1a.

Appendices

Appendix A

Entropy Conservative Inviscid Burgers' Equation

The following entropy conservative inviscid Burgers' equation was presented earlier in Tadmor [25]. Burgers' equation, which includes an artificial diffusion in flux function as in equation (2.40)

$$v_t + \left(\frac{v^2}{2}\right)_x = 0. \tag{A.1}$$

To add artificial numerical viscosity in the entropy conservative flux, the $\frac{1}{3}$'s trick is used as in [25]. The purpose to apply this is to obtain a quadratic entropy conservative flux and demonstrate the application of Tadmor's comparison principle. Then equation (A.1) becomes

$$v_t + \left(\frac{v^2}{3} + \frac{v^2}{6}\right)_x = 0$$

$$v_t + \left(\frac{v^2}{3}\right)_x + \frac{v}{3}v_x = 0 \tag{A.2}$$

Then semi-discrete form of equation (A.2) is

$$(v_j)_t + \frac{1}{3}\left(\frac{(v_{j+1})^2 - (v_{j-1})^2}{2\Delta x}\right) + \frac{1}{3}v_j\left(\frac{v_{j+1} - v_{j-1}}{2\Delta x}\right) = 0,$$

where Δx is the mesh size between two consecutive grid points. Then the same $\frac{1}{3}$'s trick is used once again to obtain numerical entropy conservative fluxes with symbolic

artificial diffusion

$$
\begin{aligned}
&(v_j)_t + \frac{1}{2\Delta x}\left(\frac{1}{2}(v_{j+1})^2 - \frac{1}{2}(v_{j-1})^2\right) - \frac{1}{2\Delta x}\left(\frac{1}{6}(v_{j+1})^2 - \frac{1}{6}(v_{j-1})^2\right) \\
&\qquad + \frac{1}{2\Delta x} v_j \left(\frac{v_{j+1} - v_{j-1}}{3}\right) = 0 \\
&(v_j)_t + \frac{1}{2\Delta x}\left(\frac{1}{2}(v_{j+1})^2 - \frac{1}{2}(v_{j-1})^2\right) - \frac{1}{2\Delta x}\frac{1}{6}\left((v_{j+1} - v_j)^2 - (v_j + v_{j-1})^2\right) = 0 \\
&(v_j)_t + \frac{1}{2\Delta x}\left(\frac{1}{2}(v_{j+1})^2 - \frac{1}{2}(v_{j-1})^2\right) - \frac{1}{2\Delta x}\left(\frac{1}{6}(\Delta v_{j+1/2})^2 - \frac{1}{6}(\Delta v_{j-1/2})^2\right) = 0
\end{aligned}
$$

where $\Delta v_{j+1/2} = v_{j+1} - v_j$ and $\Delta v_{j-1/2} = v_j - v_{j-1}$. Then the entropy conservative flux is

$$\hat{g}_{i+1/2} = \frac{v_i^2 + v_{i+1}^2}{4} - \frac{v_{i+1} - v_i}{12}. \tag{A.3}$$

Neglecting the negative artificial diffusion parts, we obtain the least diffusive entropy stable Burger's equation [25] as in equation (2.107) of Chapter 2,

$$g^{es}_{i+1/2} = \frac{v_i^2 + v_{i+1}^2}{4} - \frac{(v_{i+1} - v_i)^+}{12}. \tag{A.4}$$

Appendix B

The Algorithm of Logarithmic Mean

In [68], the logarithmic mean stated in equation (2.116) is defined in a different way as follows

$$(m_1, m_2)^{\ln} = \frac{m_1 + m_2}{\ln \varsigma} \frac{\varsigma - 1}{\varsigma + 1}, \tag{B.1}$$

where $\varsigma = m_1/m_2$ and

$$\ln(\varsigma) = 2\left(\frac{1-\varsigma}{1+\varsigma} + \frac{1}{3}\frac{(1-\varsigma)^3}{(1+\varsigma)^3} + \frac{1}{5}\frac{(1-\varsigma)^5}{(1+\varsigma)^5} + \frac{1}{7}\frac{(1-\varsigma)^7}{(1+\varsigma)^7} + O(\varsigma^9)\right). \tag{B.2}$$

The reason to redefine the logarithmic mean in equation (2.116) as in equation (B.1) is to prevent any possible non well-posed numerical solutions when $m_1 \to m_2$. Therefore, the following subroutine is used to calculate logarithmic mean (B.1) as in [68]:

1) Let $\varsigma = \frac{m_1}{m_2}$, $k = \frac{\varsigma - 1}{\varsigma + 1}$, $u = k * k$,

2) $F = \begin{cases} 1.0 + u/3.0 + u * u/5.0 + u * u * u/7.0, & \text{if } u < \epsilon \\ \frac{\ln(\varsigma)}{2.0} k, & \text{otherwise,} \end{cases}$

3) $(m_1, m_2)^{\ln} = \frac{m_1 + m_2}{2F}$ with $\epsilon = 10^{-2}$.

Appendix C

Matlab Code for position angle of shock waves

This piece of Matlab code has been used to determine the angle of the traveling shock waves at the inner and outer walls :

```
% inner wall derivative of density with respect to theta
drhodtheta1 = (X(1, 2:end, 1) - X(1, 1:end-1, 1))/dtheta;
a1 = abs(drhodtheta1);
% outer wall derivative of density with respect to theta
drhodtheta2 = (X(N, 2:end, 1) - X(N, 1:end-1, 1))/dtheta;
a2 = abs(drhodtheta2);
% to find the place of shock wave
[C1, inner_th] = max(a1);
[C2, outer_th] = max(a2);
% shock positions at inner wall
shock_theta_inner(k) = th_c_2(1, inner_th);
% shock positions at outer wall
shock_theta_outer(k) = th_c_2(N, outer_th);
```

Bibliography

[1] Anderson, J. D. (1990). Modern compressible flow: with historical perspective. *McGraw Hill Higher Education*.

[2] Jenkins, D. R., Landis, T., & Miller, J. (2003). American X-vehicles: An inventory, X-1 to X-50. *National Aeronautics and Space Administration, Office of External Relations, History Office*.

[3] Minzoni, A. A., & Smyth, N. F. (2015). Gerald Beresford Whitham. 13 December 1927—26 January 2014. *Biographical Memoirs of Fellows of the Royal Society*.

[4] Liepmann, H. W., & Roshko, A. (1957). Elements of gasdynamics. *Courier Corporation*.

[5] Courant, R., Friedrichs, K.O., & Lewy, H. (1967). On the partial difference equations of mathematical physics. *IBMJ*, 11, 215-234.

[6] Finn, B. S. (1964). Laplace and the Speed of Sound. *ISIS*, 55(179), 7.

[7] Mach, E. (1887). Photographische Fixierung der durch Projektile in der Luft eingeleiten Vorgange. *Presented to the Academy of Sciences*, Vienna

[8] Mack, Pamela. E. (1998). From engineering science to big science: the NACA and NASA Collier Trophy research project winners. *Government Printing Office*.

[9] Flax, A. (1996). National Academy of Engineering of the USA, Memorial Tributes, Vol. 8, Jakob Ackeret.

[10] Whitham, G.B. (1974). Linear and Nonlinear Waves. *John Wiley & Sons, New York*.

[11] Hugoniot, P. H. (1887). Mèmoire sur la propagation du movement dans les corps et plus spècialement dans les gaz parfaits, *le Partie, J. Ecole Polytech. Paris*, 57, 3–97.

[12] Rankine, W. M. (1870). On the thermodynamic theory of waves of finite longitudinal disturbance. *Philosophical Transactions of the Royal Society of London*, 277-288.

[13] Mattsson, A. E., & Rider, W. J. (2015). Artificial viscosity: back to the basics. *International Journal for Numerical Methods in Fluids*, 77(7), 400-417.

[14] VonNeumann, J., & Richtmyer, R. D. (1950). A method for the numerical calculation of hydrodynamic shocks. *Journal of applied physics*, 21(3), 232-237.

[15] LeVeque, R.J. (1992). Numerical methods for conservation laws. *ETH Zurich*.

[16] Richardson, L. F. (1911). The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 307-357.

[17] Fisher, T. C. (2012). High-order L2 stable multi-domain finite difference method for compressible flows. *Doctoral dissertation, Purdue University*.

[18] Hirsch, C. (2002). Numerical computation of internal and external flows: volume 2: computational methods for inviscid and viscous flows. *John Wiley & Sons*.

[19] Runge, C. (1895). Über die numerische Auflösung von Differentialgleichungen. *Mathematische Annalen*, 46(2), 167-178.

[20] Heun, K. (1900). Neue Methoden zur approximativen Integration der Differentialgleichungen einer unabhängigen Veränderlichen. *Z. Math. Phys*, 45, 23-38.

[21] Kutta, W. (1901). Beitrag zur näherungweisen Integration totaler Differentialgleichungen.*Zeitschr. für Math. u. Phys.*, 46, 435-453

[22] Lax, P. D. (1967). Hyperbolic difference equations: A review of the Courant-Friedrichs-Lewy paper in the light of recent developments. *IBM Journal of Research and Development*, 11(2), 235-238.

[23] Osher, S. (1984). Riemann solvers, the entropy condition, and difference approximations. *SIAM J. Numer. Anal.* 21, 217-235.

[24] Tadmor, E. (1987). The numerical viscosity of entropy stable schemes for systems of conservation laws. I. *Mathematics of Computation*, 49(179), 91-103.

[25] Tadmor, E. (2003). Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems. *Acta Numerica*, 12, 451-512.

[26] Godunov, S. K. (1961). An interesting class of quasilinear systems, *DAN, USSR*, 139(3), 521-523.

[27] Mock, M. S. (1980). Systems of conservation laws of mixed type. *Journal of Differential equations*, 37(1), 70-88.

[28] Harten, A. (1983). On the symmetric form of systems of conservation laws with entropy. *Journal of computational physics*, 49(1), 151-164.

[29] Svärd, M. & Özcan, H. (2014). Entropy-stable schemes for the Euler equations with far-field and wall boundary conditions. *Journal of Scientific Computing,* **58**, 61–89.

[30] Hughes, T. J., Franca, L. P., & Mallet, M. (1986). A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. *Computer Methods in Applied Mechanics and Engineering*, 54(2), 223-234.

[31] Dafermos, C.M. (2000). Hyperbolic conservation laws in continuum physics. *Springer, Berlin.*

[32] Lax, P. D., & Richtmyer, R. D. (1956). Survey of the stability of linear finite difference equations. *Communications on Pure and Applied Mathematics*, 9(2), 267-293.

[33] Lax, P., & Wendroff, B. (1958). Systems of conservation laws. *Technical Report LA-2285, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, USA*.

[34] Lax, P., & Wendroff, B. (1960). Systems of conservation laws. *Communications on Pure and Applied mathematics*, 13(2), 217-237.

[35] Svärd, M., & Mishra, S. (2012). Entropy stable schemes for initial-boundary-value conservation laws. *Zeitschrift für angewandte Mathematik und Physik*, 63(6), 985-1003.

[36] Olsson, P., & Oliger, J. (1994). Energy and maximum norm estimates for nonlinear conservation laws. *Research Institute for Advanced Computer Science, NASA Ames Research Center*.

[37] Dubois, F., & Le Floch, P. (1989). Boundary conditions for nonlinear hyperbolic systems of conservation laws. *Vieweg + Teubner Verlag*, 96-104.

[38] Fjordholm, U. S., Mishra, S., & Tadmor, E. (2012). Arbitrarily high-order accurate entropy stable essentially nonoscillatory schemes for systems of conservation laws. *SIAM Journal on Numerical Analysis*, 50(2), 544-573.

[39] Tadmor, E., & Zhong, W. (2006). Entropy stable approximations of Navier–Stokes equations with no artificial numerical viscosity. *Journal of Hyperbolic Differential Equations*, 3(3), 529-559.

[40] Dutt, P. (1988). Stable boundary conditions and difference schemes for Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 25(2), 245-267.

[41] Svärd, M. (2012). Third-order accurate entropy-stable schemes for initial-boundary-value conservation laws. *Zeitschrift für angewandte Mathematik und Physik*, 63(4), 599-623.

[42] Edwards, D.H., Fearnley, P., & Nettleton, M.A. (1983). Shock diffraction in channels with 90° bends. *J. Fluid Mech.*, 132, 257-270.

[43] Henshaw, W.D., Smyth, N.F., & Schwendeman, D.W. (1986). Numerical shock propagation using geometrical shock dynamics. *J. Fluid Mech.* 171, 519-545.

[44] Courant, R., & Friedrichs, K. O. (1977). Supersonic flow and shock waves. *Springer Science & Business Media.*

[45] Schwendeman, D. W. (1993). A new numerical method for shock wave propagation based on geometrical shock dynamics. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 441(1912), 331-341.

[46] Schwendeman, D. W. (1988). A numerical scheme for shock propagation in three dimensions. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 416(1850), 179-198.

[47] Whitham, G. B. (1957). A new approach to problems of shock dynamics Part I Two-dimensional problems. *Journal of Fluid Mechanics*, 2(2), 145-171.

[48] Whitham, G. B. (1959). A new approach to problems of shock dynamics Part 2. Three-dimensional problems. *Journal of Fluid Mechanics*, 5(3), 369-386.

[49] Gottlieb, S., Shu, C.W., & Tadmor, E. (2001). Strong stability preserving high order time discretization methods. *Siam Review*, 43(1), 89-112.

[50] LeVeque, R.J. (2002). Finite volume methods for hyperbolic problems. *Cambridge University Press.*

[51] Boehm, B.W.(1984). Verifying and validating software requirements and design specifications. *IEEE Software* 1(1), 75-88.

[52] Roache, P.J. (1999). Fundamentals of computational fluid dynamics. *Hermosa Publishers, Albuquerque, NM* Chapter 18.

[53] Roache, P.J. (1998). Verification and validation in computational science and engineering. *Hermosa Publishers, Albuquerque, NM.*

[54] Sod, G.A. (1978). A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.* 27, 1-31.

[55] Tadmor, E. (1984). Numerical viscosity and the entropy condition for conservative difference schemes. *Mathematics of Computation*, 43(168), 369-381.

[56] Lax, P. D. (1973). Hyperbolic systems of conservation laws and the mathematical theory of shock waves. *SIAM*.

[57] Merriam, M. L. (1989). An entropy-based approach to nonlinear stability. *NASA Technical Memorandum*, 101086(64), 1.

[58] Smoller, J. (1994). Shock waves and reaction—diffusion equations *Springer Science & Business Media*, Vol. 258.

[59] Levandosky, J. (2002). Conservation Laws [Lecture Notes]. *Retrieved from http://web.stanford.edu/class/math220a/handouts/conservation.pdf*

[60] Friedrichs, K. O., & Lax, P. D. (1971). Systems of conservation equations with a convex extension. *Proceedings of the National Academy of Sciences*, 68(8), 1686-1688.

[61] Goodman, J., & Lax, P. D. (1988). On dispersive difference schemes. I. *Communications on pure and applied mathematics*, 41(5), 591-613.

[62] Harten, A., & Lax, P. D. (1981). A random choice finite difference scheme for hyperbolic conservation laws. *SIAM Journal on Numerical Analysis*, 18(2), 289-315.

[63] Roe, P. L. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2), 357-372.

[64] Strang, G. (1964). Accurate partial difference methods. *Numerische Mathematik*, 6(1), 37-46.

[65] Nordström, J., & Carpenter, M. H. (1999). Boundary and interface conditions for high-order finite-difference methods applied to the Euler and Navier–Stokes equations. *Journal of Computational Physics*, 148(2), 621-645.

[66] Svärd, M., Carpenter, M. H., & Nordström, J. (2007). A stable high-order finite difference scheme for the compressible Navier–Stokes equations, far-field boundary conditions. *Journal of Computational Physics*, 225(1), 1020-1038.

[67] Burgers, J. M. (1948). A mathematical model illustrating the theory of turbulence. *Advances in applied mechanics*, 1, 171-199.

[68] Ismail, F., & Roe, P. L. (2009). Affordable, entropy-consistent Euler flux functions II: Entropy production at shocks. *Journal of Computational Physics*, 228(15), 5410-5436.

[69] Barth, T. (1999). An introduction to recent developments in theory and numerics of conservation laws. *In Numerical Methods for Gasdynamic Systems On Unstructured Meshes*.

[70] Roe, P.L. Affordable, entropy-consistent, Euler flux functions I. Analytical results, *J. Comput. Physics*, submitted for publication.

[71] Laney, C. B. (1998). Computational gasdynamics. *Cambridge University Press*.

[72] Zaide, D. W., & Roe, P. L. (2009). Entropy-based mesh refinement, II: a new approach to mesh movement. *19th AIAA computational fluid dynamics*.

[73] Harten, A., Lax, P. D., & Leer, B. V. (1983). On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM review*, 25(1), 35-61.

[74] Shu, C. W., & Osher, S. (1989). Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *Journal of Computational Physics*, 83(1), 32-78.

[75] Woodward, P., & Colella, P. (1984). The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54(1), 115-173.

[76] Roache, P.J. (2001). Code verification by the method of manufactured solutions. *J. Fluids. Eng.* 124(1), 4-10.

[77] Bond, R.B., Ober, C.C., & Knupp, P.M. (2006). A manufactured solution for verifying CFD boundary conditions, part III. *36th AIAA Fluid Dynamics Conference* 3, 1966-1982.

[78] Roy, C.J., Nelson, C.C., Smith, T.M., & Ober, C.C. (2004). Verification of Euler/Navier-Stokes codes using the method of manufactured solutions. *Int. J. Numer. Meth. Fluids* 44, 599-620.

[79] Brunner, T.A. (2006). Development of a grey nonlinear thermal radiation diffusion verification problem. *Transactions of the American Nuclear Society* 95, 876-878.

[80] Domino, S.P., Wagner, G., Hanlin, A.L., Black, A., & Land, J.S. (2007). Verification of multi-mechanics applications. *48th AIAA/ASME/ASCE/ASC Structures, Structural Dynamics, and Materials Conference.*

[81] Tremblay, D., Etienne, S., & Pelletier, D. (2006). Code verification and the method of manufactured solutions for fluid-structure interaction problems. *36th AIAA Fluid Dynamics Conference* 2, 882-892.

[82] Vedovoto, J.M., Neto, A.S., Mura, A., & Silva, L.F.F. (2011). Application of the method of manufactured solutions to the verification of a pressure-based finite-volume numerical scheme. *Computers and Fluids* 51(1), 85-99.

[83] Veluri, S.P. Code verification and numerical accuracy assessment for finite volume CFD codes. *Ph.D. Thesis.*

[84] Danaila, I., Joly, P., Kaber, S.M., & Postel, M. (2007). An introduction to scientific computing twelve computational projects solved with Matlab. *Sppringer Science + Business Media, LLC.*

[85] Jameson, A., Schmidt, W., & Turkel, E. (1981). Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes. *AIAA* pap 81-1259.

[86] Shu, C.W., & Osher, S. (1988). Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, 77, 439-471.

[87] Gottlieb, S., & Shu, C.W. (1998). Total variation diminishing Runge-Kutta schemes. *Math. Comp.*, 67, 73-85.

[88] Rezzolla, L., & Zanotti, O. (2013). Relativistic hydrodynamics. *Oxford University Press.*

[89] Tsangaris, S., & Pappou, T. Analytical solutions for the unsteady compressible flow equations serving as test cases for the verification of numerical schemes. *Laboratory of Aerodynamics, National Technical University of Athens.*

[90] Batchelor, G.K. (2002). An introduction to fluid dynamics. *Cambridge University Press.*

[91] Davis, S.S. (2002). A Note on Conservation Laws for Keplerian Flows. *The Astrophysical Journal Supplement Series* 140(2), 581-588.

[92] Hirsch, C. (2007). Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics (Vol. 1). *Butterworth-Heinemann.*

[93] Mitchell M. Engauge Digitizer 2.14-digitizing software. `http://digitizer.sourceforge.net/`.